

# **Rutas seguras en la ciudad para discapacitados psíquicos**

**Laura Aragón Tebar  
Mario Arranz Roperó  
Sergio Cabrerizo Galán**

**PROYECTO DE FIN DE CARRERA  
FACULTAD DE INFORMÁTICA  
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN  
UNIVERSIDAD COMPLUTENSE DE MADRID**



**20 de Junio de 2014**

Directores:  
Jesús Escribano Martínez  
Victoria López López



# Autorización

Laura Aragón Tebar, Mario Arranz Roperó y Sergio Cabrerizo Galán, alumnos matriculados en la asignatura de Sistemas Informáticos, autorizan a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria como el código, los contenidos audiovisuales, la documentación y/o el prototipo desarrollado durante el curso académico 2013-2014 bajo la dirección de María Victoria López López, profesora del Departamento de Arquitectura de Computadores y Automática y Jesús Escribano Martínez, profesor del Departamento de Sistemas Informáticos y Computación.

Laura Aragón Tebar

Mario Arranz Roperó

Sergio Cabrerizo Galán



# Resumen

Con la tecnología de la que disponemos actualmente es posible crear herramientas informáticas que permitan la integración total de las personas discapacitadas en la sociedad. Ya existen aplicaciones móviles que permiten enviar y recibir avisos de emergencia, ayudar a discapacitados físicos a buscar y trazar rutas accesibles o que ayudan a trazar un camino seguro para que los niños vayan al colegio. Aun así, no existe ninguna que permita a un discapacitado psíquico valerse por sí mismo a la hora de desplazarse por su ciudad.

Con el objetivo de llenar este vacío se crea este proyecto, que surge de la colaboración de la Fundación ONCE y la Facultad de Informática de la Universidad Complutense de Madrid. **Rutas seguras en la ciudad para discapacitados psíquicos** se compone de una aplicación web llamada **WebTutor**, desde la cual se configura una araña de movilidad para la persona discapacitada, y dos aplicaciones móviles: **MiTutor**, con la cual un usuario puede acceder a una ruta previamente configurada por su tutor, y **SoyTutor**, que permite al tutor gestionar la información de los usuarios a su cargo desde su dispositivo móvil. Para la organización del proyecto se empleó un método de desarrollo ágil organizado en sprints por tareas. Se utilizaron el entorno Eclipse (junto al SDK de Android) y la API de Google Maps, Bitbucket como repositorio de código y Sourcetree para gestionarlo. Todo el proyecto fue realizado de acuerdo a unas recomendaciones de la *fundación ONCE* para que las aplicaciones puedan ser usadas por una amplia tipología de usuarios, incluyendo tutores y usuarios ciegos. De cara al futuro será necesario ampliar las arañas de movilidad y recibir “feedback” de tutores, discapacitados y centros de discapacitados en general.

# Abstract

Thanks to the level of technology we have reached nowadays, it is possible to develop computer programs that allow integration in society of disabled people. There are already mobile applications that allow sending and receiving emergency messages, which help physical, disabled to search and draw up an accessible path or that help to draw up a safe path for children to go to school. Even so, it does not exist any application that allows a mental disabled people to fend for their selves when they move through their city.

**Rutas seguras en la ciudad para discapacitados psíquicos** was created with the aim of filling this void. It is a project that born from the collaboration of *ONCE foundation* with the Computer science faculty of de University Complutense from Madrid. **Safe routes in the city to mentally handicapped** is composed of a web application called **WebTutor**, from where mobility spiders can be configured for disabled people, and two mobile applications: **MiTutor**, from where a disabled user can go into a path previously drew up by his tutor, and **SoyTutor**, that allows the tutor to manage all data about the users in his care from his mobile device. Agile development method organized in sprints was used to organize the project. It was also used Eclipse IDE (with Android SDK) and Google Maps API, Bitbucket as software repository and Sourcetree to manage it. The whole project was made according to several recommendations of *ONCE foundation*, to make applications accessible to blind users and tutors. Facing toward the future it will be necessary to increase the mobility spiders and receive feedback from tutors, users and disabled centers.

# Índice

Autorización.....	I
Resumen.....	II
Abstract .....	III
Capítulo 1: Introducción.....	1
1.1 Descripción general.....	1
1.2 Estructura de la memoria .....	2
1.3 Estado del arte.....	3
Aplicación Accessibility .....	3
Aplicación SafetyGPS V3.....	7
Aplicación Sloper.....	9
Capítulo 2: Requisitos .....	11
2.1 . Requisitos del usuario .....	11
2.2. Requisitos del sistema .....	11
Requisitos funcionales.....	11
Requisitos no funcionales.....	15
Capítulo 3: Técnicas y herramientas especiales utilizadas.....	17
Capítulo 4: Desarrollo .....	23
4.1. Evolución temporal .....	23
4.2. Servidor y base de datos.....	24
4.3. API .....	26
4.4. Arquitectura de la web (WebTutor) .....	29
Casos de uso.....	29
Paquetes y clases .....	54
4.5 Arquitectura de la Aplicación del usuario (MiTutor).....	59
Casos de Uso .....	59
Paquetes y clases .....	74
4.6 Arquitectura de la Aplicación del guía (SoyTutor).....	81
Casos de Uso .....	81
Paquetes y clases .....	94
Capítulo 5: Conclusiones y trabajos futuros .....	103
Bibliografía .....	105
Anexo 1: Definiciones, acrónimos y abreviaturas.....	107
Anexo 2: Manual de usuario .....	109
Página web (WebTutor) .....	109
Aplicación del usuario (MiTutor).....	124
Aplicación del guía (SoyTutor).....	133

Anexo 3: Registro de reuniones .....	143
Índice de figuras .....	145
Índice de tablas.....	149



# Capítulo 1: Introducción

En este capítulo introductorio se muestra la motivación de este proyecto así como el estado del arte relativo a aplicaciones desarrolladas para facilitar la movilidad de las personas discapacitadas.

## 1.1 Descripción general

**Rutas seguras en la ciudad para discapacitados psíquicos** se trata de un conjunto de aplicaciones cuyo objetivo principal es conseguir la autonomía de los discapacitados psíquicos a la hora de realizar sus desplazamientos cotidianos.

Las aplicaciones propuestas se han separado en dos tipos de usuarios: el discapacitado psíquico y el tutor. Para ello, se han desarrollado herramientas que ayudan al discapacitado a moverse sin compañía, y además permiten que su tutor tenga información en tiempo real sobre él, siendo esta información accesible desde un dispositivo móvil y configurable a través de un ordenador.

A lo largo de este documento se describe el procedimiento de realización del proyecto, así como técnicas y herramientas utilizadas para ello. Se explican a continuación las principales características de cada una de las tres aplicaciones. Siendo el objetivo de este apartado el de crear una idea general de estas, se describen las funcionalidades principales.

### Descripción general – Aplicación web para el tutor - Webtutor

**Webtutor** se trata de una aplicación web desde la cual cualquier guía, padre o tutor de una persona con dificultades a la hora de orientarse, pueda configurarle a este una *araña de movilidad* por la cual le resulte fácil transitar. En la web estos dos roles se llamarán guía y usuario, respectivamente.

En la pantalla de inicio existe la posibilidad de crear un nuevo tutor, o bien, iniciar sesión si es que ya está dado de alta.

El tutor, una vez introducidos sus datos para acceder a su cuenta, tendrá a su disposición una lista de todos los usuarios a los que esté tutorizando, pudiendo añadir nuevos, eliminar uno ya existente y ver los datos personales y de contacto de cada uno de ellos.

Dispondrá, también, de una lista de todas las rutas que ha configurado a sus diferentes usuarios, pudiendo añadir nuevas, visualizar la información de cada una de ellas, modificarlas, eliminarlas o valorarlas.

El hecho de valorar una ruta consiste en decidir una puntuación para cada tramo de esta, de acuerdo a distintos factores y diferenciando, en cada tramo o calle, las aceras derecha e izquierda. A partir de esta valoración se elaborará la ruta más adecuada para el usuario.

Se ha contemplado el caso de que uno o varios usuarios cambien de tutor. El tutor que traspasa usuarios a otro deberá exportarlos desde su cuenta, para que el que le releve, una vez iniciada sesión con sus datos de acceso, los importe. De este modo pasará, este último, a manejar toda la información recopilada hasta el momento de este o estos usuarios, y el primero dejará de visualizarlos en su cuenta.

## **Descripción general – Aplicación móvil para el usuario - MiTutor**

**MiTutor** es una aplicación móvil para dispositivos con sistema operativo Android, desde la cual un usuario puede acceder a una ruta previamente configurada por su tutor para llegar a uno de sus destinos habituales.

Tendrá la posibilidad de marcar una incidencia en la posición en la que se encuentra, que le será notificada al tutor junto con un indicativo de tipo de incidencia, para que éste pueda revisar la ruta o rutas que pasen por dicho punto, con el objetivo de evitarlo.

Además, podrá realizar una llamada a su tutor rápidamente desde el menú principal o mientras se encuentra siguiendo una ruta, en caso de emergencia, por ejemplo. También se ha desarrollado un sistema de vuelta a la ruta configurada por el tutor, por si el usuario se saliera accidentalmente de esa ruta.

## **Descripción general – Aplicación móvil para el tutor - SoyTutor**

**SoyTutor** es una aplicación móvil para dispositivos con sistema operativo Android, desde la cual un guía, padre o tutor puede gestionar la información de los usuarios que tiene a cargo, pudiendo obtener sus datos personales y de contacto, las rutas configuradas y las incidencias que hayan marcado. Además, podrá realizar una llamada rápidamente a cualquiera de sus usuarios desde el menú principal.

La función principal de esta aplicación es la de tener un acceso rápido a la información que puede gestionar más cómodamente desde WebTutor, pero como ventaja frente a esta, hemos diseñado un sistema de notificaciones que informan, al instante, al guía, padre o tutor que tenga instalada esta aplicación en su dispositivo móvil, de cualquier incidencia marcada por cualquiera de sus usuarios desde la aplicación móvil MiTutor.

## **1.2 Estructura de la memoria**

En el **capítulo 1** abordaremos los requisitos. El propósito de esta sección es describir las necesidades y dificultades que se han tenido durante el desarrollo de las aplicaciones tanto web como móviles.

En el **capítulo 2** abordaremos el desarrollo. En esta sección se indica el desarrollo temporal del proyecto, así como la organización de la base de datos, el servidor utilizado, la API propia desarrollada y la arquitectura de las aplicaciones tanto web como móviles.

En el **capítulo 3** abordaremos las técnicas y herramientas especiales utilizadas. El propósito de esta sección es analizar las técnicas utilizadas para el desarrollo del

proyecto, así como las herramientas utilizadas para repositorios, bases de datos y mapas utilizados.

En el **capítulo 4** abordaremos las técnicas, conclusiones y trabajos futuros sobre este proyecto.

## 1.3 Estado del arte

Presentamos aquellas aplicaciones o utilidades similares a este proyecto, ya sea porque nos han servido de base para mejorar ciertos aspectos, o de guía para considerar hacerlos de una u otra forma. Las más significativas se comentan a continuación.

### Aplicación Accessibility



Accessibility, una aplicación promovida por la Federación de Asociaciones de Discapacidad Física y Orgánica de la Comunidad de Madrid (FAMMA) con el apoyo de la Confederación Española de Personas con Discapacidad Física y Orgánica (COCEMFE) y Fundación Vodafone España, surge como una iniciativa para dar respuesta a las necesidades de ubicación de espacios y lugares accesibles para Personas con Movilidad Reducida, apoyándose en las nuevas tecnologías como soporte para la mejora de su integración.

Un sistema de geolocalización y realidad aumentada da información en tiempo real sobre en qué lugar y a qué distancia se encuentra el punto de interés seleccionado, que será accesible y adaptado. Entre otros puntos de interés se muestran plazas de aparcamiento reservado, cajeros automáticos, gasolineras o ambulatorios que cuenten con los mínimos de accesibilidad exigibles.

La realidad aumentada amplía la información percibida por la persona en la posición física real en la que se encuentra, mejorando la comprensión de la información.

Para ello, utiliza la información de los distintos sensores del terminal: A través de la cámara, ofrece en imágenes lo que está observando; a través del GPS, ofrece la posición exacta de la persona; a través de la brújula digital, ofrece la dirección en que se está mirando; a través del acelerómetro, ofrece el ángulo de visión. La aplicación proporciona cuatro funcionalidades como se muestra en la Ilustración 1.

Realidad  
aumentada



Consultar Puntos  
de Interés

Añadir Puntos  
de Interés



Herramientas

*Ilustración 1: Aplicación Accessibility: modos a elegir*



Ilustración 2: Aplicación Accessibility: Realidad aumentada



Ilustración 3: Aplicación Consultar puntos de interés

### **Realidad aumentada (Ilustración 2)**

A través del sistema de realidad aumentada se ofrecen hasta 15 Puntos de Interés. Mediante flechas, marca la dirección del Punto de Interés, su distancia en línea recta y el icono de la categoría a la que pertenece. Otra posible forma de visualización es mediante carteles. A través de un radar, la aplicación marca la posición de los Puntos de Interés y el icono de su categoría. También se puede acceder a los Puntos de Interés encontrados mediante mapa o lista. En este último caso, el icono de la categoría a la que pertenece el Punto de Interés cambiará su intensidad según se oriente el dispositivo, a fin de ofrecer con mayor precisión en qué dirección se encuentra.

### **Consultar puntos de interés (Ilustración 3)**

Los Puntos de Interés (POI) se ofrecen agrupados por categorías. Se puede elegir una de ellas, o que la aplicación muestre todos los POI localizados, independientemente de la categoría a la que pertenezcan. Entre otros, se muestran plazas de aparcamiento reservado, cajeros automáticos, gasolineras, centros de salud, transporte, así como ocio, turismo y comercios accesibles. Además de la información que cada ficha arroja en modo texto (dirección y descripción del Punto de Interés), la aplicación permite mediante iconos conectar con:

- Google Maps para obtener la ruta a pie, transporte público o coche.
- Llamar por teléfono, enviar un correo electrónico o utilizar el navegador para consultar la página web del punto seleccionado.
- Enviar un aviso si el Punto de Interés (POI) está mal localizado.
- Añadir el POI a los Favoritos.

### **Añadir puntos de interés (Ilustración 4)**

La aplicación incorpora un cuestionario para que sean los propios usuarios los que puedan actualizar la lista de servicios disponibles, si encuentran algún punto de interés accesible a personas con discapacidad que no esté ya geolocalizado. Estos puntos estarán disponibles cuando sean revisados y aprobados por FAMMA.

### **Herramientas (Ilustración 5)**

A través de esta pantalla se puede elegir el idioma. En la actualidad, los idiomas disponibles son el inglés y el español. Además, se puede decidir si se desea acceder sólo a los puntos proporcionados por FAMMA, o también a los reportados por los usuarios.

La información representada en la aplicación en cuanto a los Puntos de Interés se refiere ha sido gestionada y/o verificada por FAMMA en su gran mayoría, salvo:

- La información que está identificada como reportada por los usuarios y no ha sido etiquetada por FAMMA.
- La información de posicionamiento de las paradas de las líneas de la EMT (Empresa Municipal de Transportes de Madrid) que ha sido cedida para su uso, no siendo el contenido propiedad de FAMMA.

En caso de fuerte impacto del terminal o detección de deceleración brusca, la aplicación ofrecerá conectar con el 112 por si fuera necesario contactar en caso de emergencia.



¿Quieres añadir una localización?

**Categoría**  
Plazas de aparcamiento ▼

**Título**  
[Input field]

**Dirección**  
[Input field]

**Descripción**  
Por favor, indique si el lugar dispone de acceso a cota cero o baño adaptado

Enviar

*Ilustración 4: Aplicación Accessibility:  
Añadir puntos de interés*



? accessibility

[Icon with Spanish and UK flags]

Selecciona qué contenido quieres ver

FAMMA ☒

Usuarios ☒

*Ilustración 5: Aplicación Accessibility:  
Herramientas*

## Aplicación SafetyGPS V3



SafetyGPS es un sistema avanzado e inteligente para enviar información geolocalizada de forma bidireccional entre usuarios y entidades.

Es la primera APP móvil especializada en avisos de emergencia hacia las entidades y desde las entidades hacia la población. Avalada por el 112 de Madrid, Ministerio del Interior, Ministerio de Sanidad y decenas de ayuntamientos de España.

Primera APP especializada en emergencias para toda la población incluido personas con discapacidad. Incluye un interface especial para personas con discapacidad visual. Avalada en accesibilidad por la Secretaría General de Apoyo a la Discapacidad del Ministerio de Sanidad y la CNSE (Confederación estatal de persona sordas).

Su uso es gratuito tanto para usuarios como para entidades públicas españolas. Es un proyecto del informático español Eugenio Herrero y se ha desarrollado completamente en España.



*Ilustración 6: Aplicación SafetyGPS V: Pantalla principal*





*Ilustración 7: Aplicación SafetyGPS V: Funcionalidades*

- Enviar avisos de emergencias y mantenimiento urbano a las administraciones públicas como por ejemplo a las decenas de ayuntamientos que ya usan SafetyGPS.
- Recibir avisos de emergencias de administraciones públicas como por ejemplo evacuación de zonas, avisos de temporales, calles cortadas, recibir noticias locales, etc...
- Usando la entidad @Cambiaticiudad podrás compartir vía Twitter lo que te gusta, lo que odias, sugerencias y arreglos de la ciudad donde estés aunque su ayuntamiento todavía no use SafetyGPS.
- Cada día son más los ayuntamientos que usan SafetyGPS para recibir avisos sobre el estado de la ciudad así como opiniones de la población.
- Recibir información de los ayuntamientos, como localización de farmacias, noticias, ofertas de empleo y cualquier otra información que el ayuntamiento desee compartir.

Como usuario podrás:

- Ayudar a las autoridades a triangular fuegos en el monte. Tecnología pionera en el mundo.
- Alertar de sucesos de emergencia que te sucedan a ti o de los que seas testigo.
- Alertas de distancia entre tus contactos. Alzheimer, ancianos, niños, etc.
- Mensajería gratuita con tus contactos, comparte lugares, guarda y comparte posiciones de restaurantes, tiendas, etc.



## Aplicación Sloper

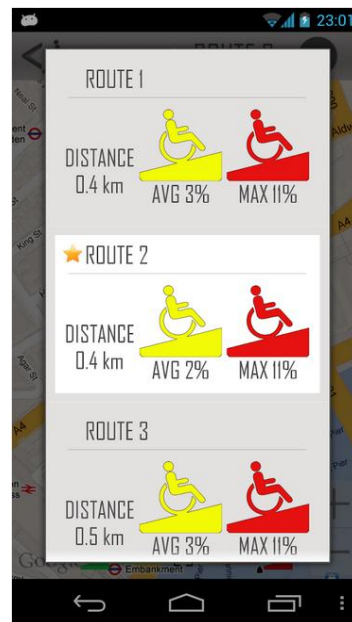
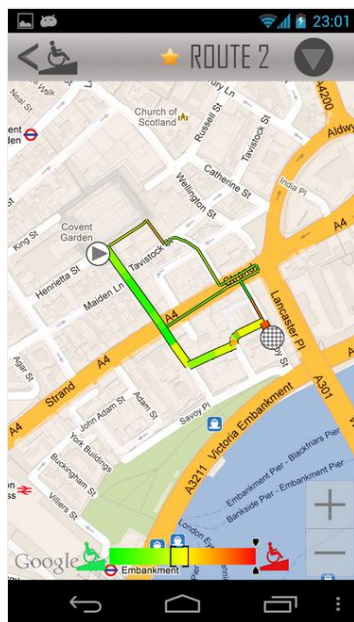


Sloper es una aplicación que intenta conseguir una nueva percepción de las calles: una forma de dibujar y leer mapas de forma nueva y revolucionaria.

Su finalidad es la de buscar rutas accesibles. Las personas con discapacidad que utilizan SLOPER podrán comprobar en el mapa si el camino que han elegido es completamente accesible o algún tipo de barrera arquitectónica se puede encontrar en su camino. La ventaja es clara: la movilidad al alcance de todos. La aplicación mostrará tres rutas y marcará la más fácil. Además, el usuario será capaz de comparar las distintas rutas: distancia total, pendiente máxima y pendiente media.

Esta aplicación representa gráficamente las pendientes entre dos puntos usando una línea de colores. Estos colores varían entre verde y naranja para pendientes entre 0% y 6%, siendo rojo para pendientes superiores al 6%.

En los mapas turísticos, por ejemplo, se representan sólo edificios representativos o jardines, por lo que todo puede parecer plano. Una gran cantidad de información se puede dibujar en mapas ordinarios: la ubicación de los lugares más frecuentados, la estructura de la ciudad, pero ninguno sobre la dificultad de una ruta.



*Ilustración 8: Aplicación Sloper: Representación de la ruta*



## Capítulo 2: Requisitos

El propósito de este capítulo es describir las necesidades y dificultades que se han tenido durante el desarrollo de las aplicaciones **MiTutor**, **SoyTutor** y **WebTutor**. Se expondrán los requisitos de usuario y de sistema de las aplicaciones anteriormente mencionadas y los problemas que se han tenido en el desarrollo y las soluciones que se han propuesto.

### 2.1. Requisitos del usuario

Los actores principales de este proyecto compuesto por las aplicaciones **MiTutor**, **SoyTutor** y **WebTutor** son un usuario con problemas a la hora de orientarse sin compañía y su padre, madre o tutor.

Se recogen las siguientes funcionalidades principales:

- Registrar e identificar un guía y/o usuario en la aplicación, permitiendo su acceso al resto de funcionalidades.
- Consultar la lista de los usuarios de un guía.
- Consultar la lista de las rutas de un guía, o de un usuario concreto.
- Consultar la información de un usuario determinado (únicamente podrá acceder a ella su tutor).
- Añadir, modificar, valorar o eliminar una ruta, tramo o usuario.
- Marcar o visualizar incidencias.
- Recibir notificaciones.

#### Requisitos:

Los nuevos tutores deben completar correctamente el formulario de registro, rellenando los campos indicados como obligatorios.

Además pueden registrar los usuarios que tengan a su cargo y configurar sus arañas de movilidad.

Un usuario debe acceder a la aplicación móvil previo inicio de sesión de su tutor.

Los usuarios deben activar los servicios de geolocalización GPS de su dispositivo para poder hacer uso de la funcionalidad principal de la aplicación, que es la de llevarle al destino seleccionado.

### 2.2. Requisitos del sistema

Cada una de las pantallas/secciones de las aplicaciones **MiTutor**, **SoyTutor** y **Webtutor** ha sido diseñadas de acuerdo a una serie de requisitos del propio sistema, de acuerdo a un orden lógico de usabilidad. Estos requisitos han sido predefinidos antes de empezar a desarrollar, y divididos en funcionales y no funcionales.

#### Requisitos funcionales

En este apartado se explican los requisitos que tienen que ver con la funcionalidad de las aplicaciones.

- **Aplicación Móvil Del Usuario (MiTutor)**

**Acceso**

Un guía deberá acceder a la aplicación móvil, mediante un inicio de sesión donde se deberán introducir los datos (correo y contraseña) especificados en el registro. Posteriormente, elegirá de una lista de usuarios de los que es responsable/guía, el que va a utilizar dicho dispositivo.

**Marcar incidencias**

La aplicación permite al usuario marcar incidencias geolocalizadas encontradas mientras recorre la ruta seleccionada o simplemente marcar la incidencia en un momento determinado sin necesidad de estar recorriendo una ruta.

**Comenzar una ruta**

La aplicación permite que el usuario elija una ruta deseada mediante una lista de las rutas que el guía inserta a través de la página web y posteriormente el usuario será guiado por la aplicación mediante GPS desde el inicio de la ruta hasta el final de la misma.

**Llamada de pánico**

La aplicación permite realizar una llamada al móvil de contacto del usuario en un momento de pánico mientras recorre una ruta.

**Volver a la ruta**

La aplicación guiará al usuario al último tramo por donde haya pasado a través de Navigator de Google Maps.

**Llamar al tutor**

La aplicación permite realizar una llamada al móvil de su guía.

**Cerrar sesión**

La aplicación permite cerrar la sesión actual.

**Configuración**

La aplicación permite configurar la aplicación.

**Enviar notificaciones**

La aplicación envía una notificación al guía cada vez que el usuario marque una incidencia.

- **Aplicación Móvil del Guía (SoyTutor)**

**Acceso**

Un guía deberá acceder a la aplicación móvil, mediante un inicio de sesión donde se deberán introducir los datos (correo y contraseña) especificados en el registro.

**Ver incidencias**

La aplicación permite al guía ver las incidencias marcadas por sus usuarios, seleccionando una de la lista de incidencias.

**Ver una ruta**

La aplicación permite que al guía visualizar las rutas de sus usuarios, seleccionando una de la lista de rutas.

**Ver datos usuarios**

La aplicación permite al guía ver la información de cada usuario que tiene a su cargo, seleccionando una de la lista de usuarios.

**Llamar usuario**

La aplicación permite al guía llamar a sus usuarios desde las ventanas de información del usuario a través del botón llamar.

**Cerrar sesión**

La aplicación permite cerrar la sesión actual.

**Configuración**

La aplicación permite configurar la aplicación.

**Recibir notificaciones**

La aplicación recibirá una notificación cada vez que el usuario marque una incidencia.

- **Página Web (Webtutor)**

**Registro**

Un guía deberá registrarse en la aplicación web, lo cual es necesario para el posterior uso de la aplicación web y la aplicación móvil del usuario y el guía.

**Acceso**

Un guía deberá acceder a la aplicación web, mediante un inicio de sesión donde se deberán introducir los datos (correo y contraseña) especificados en el registro.

**Crear el perfil de un usuario**

El guía podrá añadir nuevos usuarios en el sistema a los cuales posteriormente se le añadirán sus respectivas rutas.

**Modificar perfil del usuario**

La aplicación permitirá al guía modificar cualquiera de los datos de sus usuarios si estos cambiaran durante algún momento en el que el usuario este registrado en la aplicación.

**Eliminar perfil del usuario**

La aplicación permitirá al guía eliminar del sistema a cualquiera de sus usuarios.

**Visualizar perfil del usuario**

La aplicación le permitirá seleccionar un usuario para mostrar toda la información del mismo.

**Crear una ruta**

Donde el guía tendrá que introducir los datos de origen, destino y el usuario al que va asociado la ruta al igual que un nombre para guardar dicha ruta.

**Confirmación de datos**

Confirmar que los datos de origen y destino son correctos, tras lo cual se le montara al guía el procedimiento que deberá seguir para completar la ruta.

**Modificar ruta**

La aplicación permitirá al guía modificar los siguientes atributos de una ruta.

**Modificar el nombre de la ruta**

La aplicación permitirá al guía modificar el nombre de la ruta en el caso de que cambiará mientras la ruta este activa en el sistema.

**Modificar valoración de los tramos de la ruta**

La aplicación permitirá al guía modificar la valoración de los tramos de una ruta en caso de que estos variarán mientras la ruta está activa en el sistema.

**Eliminar ruta**

Una vez introducida al menos una ruta el guía tendrá la posibilidad de eliminar un ruta de un usuario siendo una acción que no tiene retroceso.

**Visualizar ruta**

Una vez introducida al menos una ruta el guía podrá visualizar las rutas añadidas anteriormente.

**Marcar tramos**

El guía deberá arrastrar la trayectoria hasta un punto o una intersección para después valorar el tramo una vez acabado de marcar tantos tramos como necesite la ruta.

**Valorar tramos**

El guía valorar cada uno de los tramos marcados anteriormente diferenciando en aceras y según una serie de factores que con llevaran a denotar la puntuación final de la ruta.

**Importar usuarios**

La aplicación permitirá al guía importar usuarios y añadirlos a sus usuarios.

**Exportar usuarios**

La aplicación permitirá al guía exportar los usuarios seleccionados.

**Cerrar sesión**

La aplicación permite cerrar sesión en el sistema al guía.

## Requisitos no funcionales

En este apartado se explican los requisitos que no tienen que ver con la funcionalidad de las aplicaciones.

- **Usabilidad**

La aplicación está destinada al uso por parte de discapacitados, por lo que debe ser fácil de usar.

Los menús son sencillos y con botones grandes para un mejor manejo de la aplicación.

La aplicación permite desplazarse entre interfaces haciendo uso de los botones que hay en ellas y permite en cualquier momento ir a las interfaces anteriores utilizando el botón "atrás" del dispositivo.

La aplicación está adaptada para su correcto uso por parte de discapacitados visuales salvo el caso de los mapas de Google.

- **Fiabilidad**

Las pruebas de fiabilidad han sido realizadas sobre los terminales móviles Samsung Galaxy S3, Samsung Galaxy S4, Samsung Galaxy Nexus, Samsung Galaxy Ace 2.

- **Restricciones de diseño**

**Uso del entorno de desarrollo Eclipse**

Para la implementación de la aplicación usaremos el entorno de desarrollo Eclipse, versión Kepler, junto con el SDK (Kit de Desarrollo de Software) de Android.

**Uso del SDK de Android 2.2**

Hemos desarrollado la aplicación sobre la versión 2.2 Froyo de Android (API 8), que es la mínima versión que tiene cualquier dispositivo en la actualidad.

- **Componentes usados**

**Uso de la API de Google Maps v2**

Hemos utilizado la API de Google Maps V2 para obtener la posición GPS del usuario para poderle guiar y permitirle reportar incidencias, al igual que para mostrar los mapas.





# Capítulo 3: Técnicas y herramientas especiales utilizadas

## ACCESIBILIDAD WEB

Hablar de *Accesibilidad Web* es hablar de un acceso universal a la Web, independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios.

La idea principal radica en hacer la Web más accesible para todos los usuarios independientemente de las circunstancias y los dispositivos involucrados a la hora de acceder a la información. Partiendo de esta idea, una página accesible lo sería tanto para una persona con discapacidad, como para cualquier otra persona que se encuentre bajo circunstancias externas que dificulten su acceso a la información (en caso de ruidos externos, en situaciones donde nuestra atención visual y auditiva no están disponibles, pantallas con visibilidad reducida, etc.).

Para hacer el contenido Web accesible, se han desarrollado las denominadas Pautas de Accesibilidad al Contenido en la Web (WCAG), cuya función principal es guiar el diseño de páginas Web hacia un diseño accesible, reduciendo de esta forma barreras a la información. WCAG consiste en unas pautas que proporcionan soluciones de diseño y que utilizan como ejemplo situaciones comunes en las que el diseño de una página puede producir problemas de acceso a la información. Las Pautas contienen además una serie de puntos de verificación que ayudan a detectar posibles errores.

Cada punto de verificación está asignado a uno de los tres niveles de prioridad establecidos por las pautas.

- **Prioridad 1:** son aquellos puntos que un desarrollador Web tiene que cumplir ya que, de otra manera, ciertos grupos de usuarios no podrían acceder a la información del sitio Web.
- **Prioridad 2:** son aquellos puntos que un desarrollador Web debería cumplir ya que, si no fuese así, sería muy difícil acceder a la información para ciertos grupos de usuarios.
- **Prioridad 3:** son aquellos puntos que un desarrollador Web debería cumplir ya que, de otra forma, algunos usuarios experimentarían ciertas dificultades para acceder a la información.

En función a estos puntos de verificación se establecen los niveles de conformidad:

- **Nivel de Conformidad "A":** todos los puntos de verificación de prioridad 1 se satisfacen.
- **Nivel de Conformidad "Doble A":** todos los puntos de verificación de prioridad 1 y 2 se satisfacen.
- **Nivel de Conformidad "Triple A":** todos los puntos de verificación de prioridad 1, 2 y 3 se satisfacen.

## PAUTAS Y PUNTOS DE COMPROBACIÓN

- **Imágenes y animaciones:** Use el atributo alt para describir la función de cada elemento visual.
- **Mapas de imagen:** Use el elemento map y texto para las zonas activas.
- **Multimedia:** Proporcione subtítulos y transcripción del sonido, y descripción del vídeo.
- **Enlaces de hipertexto:** Use texto que tenga sentido leído fuera de contexto. Por ejemplo, evite "pincha aquí".
- **Organización de las páginas:** Use encabezados, listas y estructura consistente. Use CSS para la maquetación donde sea posible.
- **Figuras y diagramas:** Descríbalos brevemente en la página o use el atributo longdesc.
- **Scripts, applets y plug-ins:** Ofrezca contenido alternativo si las funciones nuevas no son accesibles.
- **Marcos:** Use el elemento noframes y títulos con sentido.
- **Tablas:** Facilite la lectura línea a línea. Resuma.
- **Revise su trabajo:** Verifique. Use las herramientas, puntos de comprobación y pautas de <http://www.w3.org/TR/WCAG>.

## AYUDAS TÉCNICAS

Para desarrollar aplicaciones accesibles para Android es necesario conocer las ayudas técnicas (ATs) que posibilitan la accesibilidad del dispositivo. A continuación se realiza un breve recorrido sobre el estado del arte de las ayudas técnicas más utilizadas y reconocidas por usuarios en la plataforma Android.

Las ATs en Android se pueden dividir en servicios de accesibilidad o métodos de entrada.

Un Servicio de Accesibilidad es un proceso que corre en segundo plano y gestiona los eventos de accesibilidad generados por el sistema. Los eventos de accesibilidad se generan cuando el usuario interactúa con la interfaz de usuario, como cuando se hace click en un botón o cuando un View tiene el foco. El objetivo de un Servicio de Accesibilidad es ofrecer un feedback alternativo al usuario. El más utilizado en Android es el lector de pantallas Talkback.

Un Método de Entrada (Input Method, IME) es una herramienta software que permite a los usuarios introducir texto y navegar por la interfaz de usuario. Se instala en el sistema Android como una aplicación más, pero ésta debe contener una clase que extienda de InputMethodService. Por lo general, estas ATs son herramientas de software que facilitan el acceso a dispositivos móviles para las personas con problemas de movilidad.

A continuación se van a describir algunas de las ayudas técnicas más importantes y utilizadas en Android:

- **Talkback:** es un Servicio de Accesibilidad oficial de Google, es el lector de pantalla más utilizado en Android. Existen dos aplicaciones de Talkback diferentes:  
Anterior a Android 4.0 (ICS). También conocido como Talkback preICS, éste es un lector de pantalla bastante simple, no tiene opciones de configuración y únicamente se limita a leer el elemento que contiene el foco.

Desde Android 4.0 (ICS) A partir de ICS todos los dispositivos incorporan el Talkback de fábrica, aquí el Talkback no es sólo un lector de pantalla, sino un servicio de accesibilidad que proporciona un feedback de todo tipo. Además, escucha eventos de onHover y offHover generados al tener activo el modo de exploración táctil. En este Talkback se permiten varios parámetros de configuración, como poder cambiar el volumen o configurar otros tipos de feedback como la vibración.

Cabe destacar que en Talkback la versión utilizada también importa en ciertos casos. Se recomienda utilizar siempre la última versión disponible, no obstante a la hora de desarrollar, debemos tener en cuenta las versiones anteriores. Un ejemplo de estas diferencias se encuentra a partir de la versión 3.3, donde ya se incluye por defecto, en los botones, el tipo de elemento que son, mientras que en las versiones anteriores, si el desarrollador no lo indicaba, sólo reproducía el nombre de la etiqueta.

- **KickBack:** es uno de los servicios de accesibilidad oficiales de Google. Provee un feedback de vibración cuando sucede un evento, como desbloquear la pantalla o cambiar el foco del elemento.  
A partir de Android 4.0 (ICS) estas funcionalidades están integradas en el Talkback.
- **SoundBack:** es uno de los servicios de accesibilidad oficiales de Google. Genera un breve sonido cuando se realiza una acción sobre el teléfono como hacer click en un elemento o mover la selección con el manejador de selección. A partir de Android 4.0 (ICS) estas funcionalidades están integradas en el Talkback.
- **Tecla Access:** es un método de entrada alternativo diseñado para facilitar el manejo del dispositivo para personas con movilidad reducida. Tecla soporta el control del móvil mediante dispositivos inalámbricos, como un teclado físico, un pulsador externo o la propia silla de ruedas del usuario. A parte del teclado normal para la introducción de texto, Tecla proporciona un teclado táctil para navegar en el móvil. También dispone de un modo en pantalla completa que permite controlar el dispositivo e introducir texto tan solo pulsando en la pantalla mediante un sistema de barrido.
- **Dasher:** es un método de entrada diseñado para la introducción de texto mediante una serie de suaves movimiento de un puntero, de esta forma se puede escribir arrastrando suavemente el dedo por la pantalla, girando el móvil a través de la información que proporciona el acelerómetro o mediante dispositivos externos como joysticks, trackballs, pulsadores, pero también punteros controlados por los pies, la cabeza, con los ojos o incluso con el Wii-mote. Dasher está planteado principalmente para usuarios con movilidad reducida, pero también está destinado a personas que simplemente utilizan este teclado por la rapidez que se puede alcanzar escribiendo.
- **EyesFree Keyboard:** es un método de entrada de texto que además proporciona una interfaz para navegar por la pantalla mediante gestos, consiste en un Controlador Direccional (D-pad) que permite mover el foco en las cuatro direcciones. Se debe utilizar junto con un lector de pantalla, recomendablemente Talkback. El teclado de texto se puede explorar con el dedo mientras se reproduce letra a letra y dando la posibilidad de escribir a personas con discapacidad visual.

No todas las interfaces se pueden explorar con el D-pad y la exploración del teclado no funciona correctamente en todos los modelos de dispositivos.

- **BrailleBack:** es una App desarrollada por Eyes-Free project, que permite conectar un dispositivo externo de braille permitiendo navegar y escribir texto mediante este método.

## ACCESIBILIDAD EN ANDROID

Las aplicaciones desarrolladas para Android son más accesibles para usuarios con limitaciones visuales, físicas o relacionadas con la edad cuando éstos activan los servicios y funcionalidades de accesibilidad en sus dispositivos, sin necesidad de cambio alguno en el código de la aplicación para adecuarla a sus necesidades. Sin embargo, es muy conveniente completar una serie de pasos para optimizar la accesibilidad de la aplicación y asegurar una experiencia agradable al usuario.

Asegurarse de que nuestra aplicación es accesible para todos los usuarios requiere tan sólo de unos pocos pasos, particularmente cuando creamos las interfaces de usuario con componentes proporcionados por el framework de Android. Si usamos exclusivamente los componentes estándar, los pasos a seguir serían los siguientes:

- Añadir textos descriptivos a las interfaces de control de la aplicación usando el atributo “android:contentDescription”. En especial en los elementos de interfaz ImageButton, ImageView y CheckBox.
- Asegurarse de que todos los elementos de la interfaz que aceptan un método de entrada (touch o click) pueden ser seleccionados mediante un control direccional como un trackball, D-pad, o gestos de navegación.
- Estar seguros de que los mensajes de audio van siempre acompañados por otro indicador visual o notificación, para ayudar a los usuarios sordos o con problemas de audición.
- Probar la aplicación utilizando únicamente los servicios de navegación y características de accesibilidad. Activar el servicio TalkBack y explorar con el tacto, y luego tratar de usar la aplicación utilizando sólo los controles direccionales.
- En caso de implementar nuestros propios controles personalizados que amplían la clase View, se debe realizar un trabajo adicional que asegure que sus componentes son accesibles. A continuación se describe el procedimiento a seguir para desarrollar componentes personalizados compatibles con los servicios de accesibilidad de Android.

## **ACCESIBILIDAD PROYECTO**

Hemos realizado el proyecto basándonos en las pautas anteriormente contadas para realizar tanto la aplicación web como las aplicaciones móviles con la mayor accesibilidad posible, ya que hay partes en las aplicaciones tanto web como la de los móviles que no hemos podido hacer accesibles para todo tipo de usuarios, es el caso del tema de los mapas de Google Maps, los cuales no hemos tenido forma de hacerlos accesibles para todos, debido a esto, esta funcionalidad solo la podrá realizar sin inconvenientes aquellas personas que no tengan discapacidad visual, ya que necesitara poder ver los mapas para poder seguir las rutas.

Tanto en las aplicaciones web y móviles, la accesibilidad a sido comprobadas mediante distintos programas, como es el caso de la aplicación web que ha sido probada mediante el validador de webs de W3C con el cual obtenemos que no tenemos ningun error de accesibilidad y por el lector de pantalla, asi como en el caso de las aplicaciones móviles ha sido comprobada mediante Talkback.



# Capítulo 4: Desarrollo

En esta sección se indica el desarrollo temporal del proyecto, así como la organización de la base de datos, el servidor utilizado y API propia desarrollada. Además se incluye un análisis de la arquitectura, englobando los casos de uso de las tres aplicaciones y la distribución de paquetes y clases usada.

## 4.1. Evolución temporal

### Tarea

Nombre	Fecha de inicio	Fecha de fin
Tormenta de ideas	1/10/13	30/10/13
Investigación de posibles tecnologías	31/10/13	29/11/13
Desarrollo	25/12/13	4/04/14
Aplicación web I	25/12/13	21/01/14
Aplicación web II	22/01/14	25/02/14
Aplicación móvil Usuario I	6/01/14	21/01/14
Aplicación móvil Usuario II	26/02/14	4/04/14
Aplicación móvil Tutor I	6/01/14	27/01/14
Aplicación móvil Tutor II	26/02/14	4/04/14
API interacción base de datos I	25/12/13	5/01/14
API interacción base de datos II	6/01/14	27/01/14
API interacción base de datos III	28/01/14	25/02/14
API interacción base de datos IV	5/04/14	25/04/14
Google Maps	22/01/14	25/02/14
Documentación (primera versión)	30/11/13	24/12/13
Prototipos	30/11/13	24/12/13
Documentación I	25/12/13	21/01/14
Documentación II	25/12/13	21/01/14
Documentación III	20/03/14	4/04/14
Organización y base de datos I	25/12/13	5/01/14
Organización y base de datos II	26/02/14	19/03/14
Diseño I	6/01/14	21/01/14
Diseño II	22/01/14	25/02/14
Diseño III	20/03/14	25/04/14
Documentación Final	26/04/14	21/05/14
Aplicación web Final	26/04/14	21/05/14
Aplicación móvil Usuario Final	26/04/14	21/05/14
Aplicación móvil Tutor Final	26/04/14	21/05/14

Ilustración 9: Diagrama de tareas

### Diagrama de Gantt

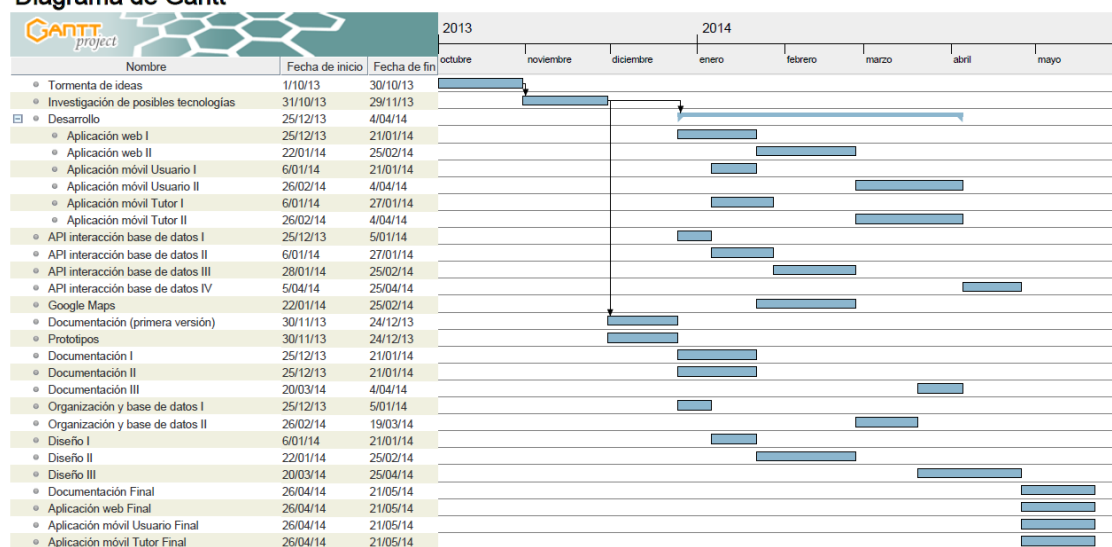


Ilustración 10: Diagrama de Gantt

## Diagrama de recursos

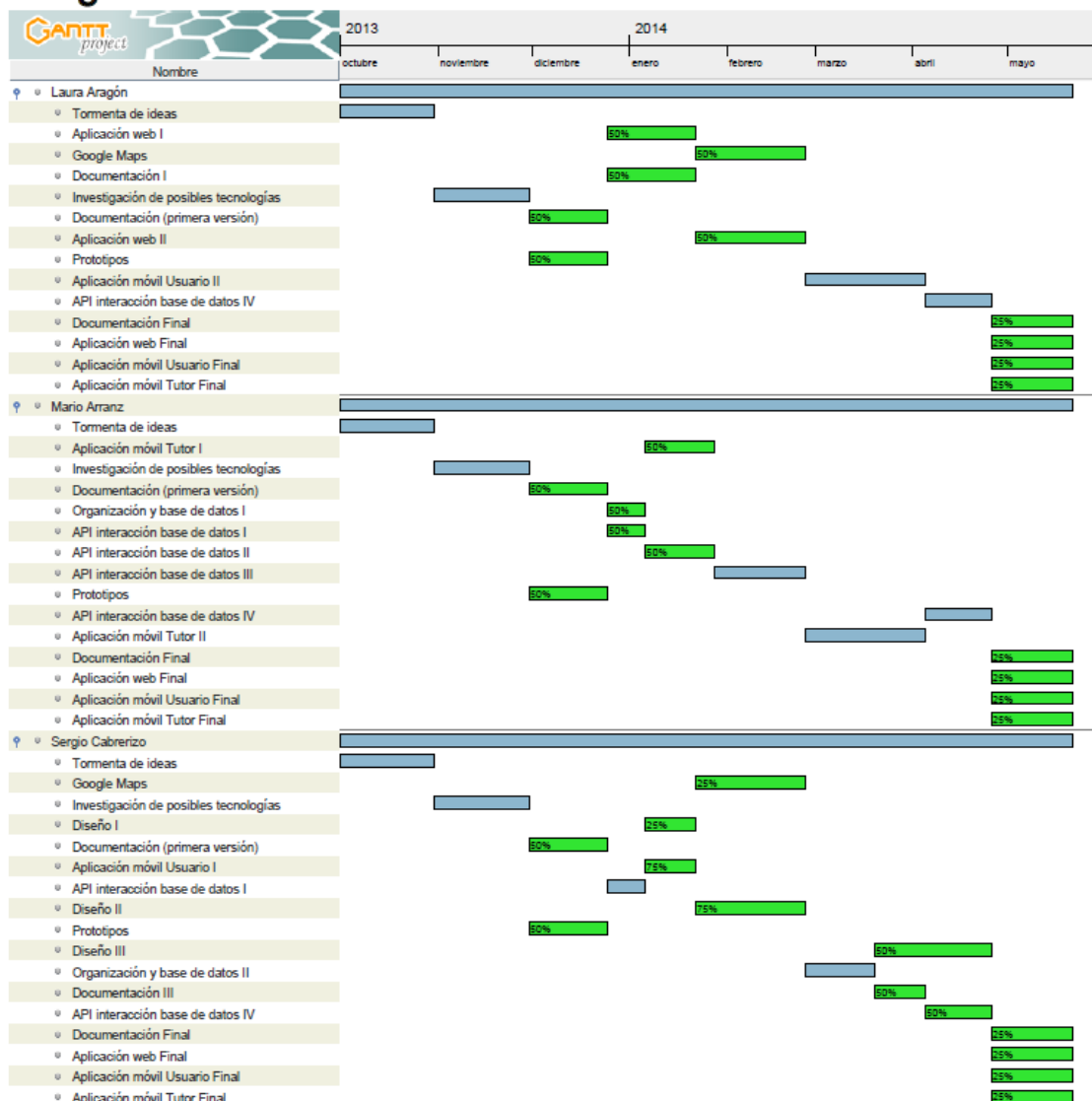


Ilustración 11: Diagrama de recursos

## 4.2. Servidor y base de datos

### SERVIDOR

El *hosting* elegido finalmente ha sido el que ofrece gratuitamente la web [hostinger.com](http://hostinger.com), que ha sido el que mayor estabilidad y rapidez de respuesta ha resultado tener.

Hemos gestionado los archivos alojados en este mediante conexión FTP, a través de la aplicación Filezilla Client.

### BASE DE DATOS

Hemos estructurado la base de datos que usarán las 3 aplicaciones con tablas distribuidas siguiendo una misma estructura:

Una tabla para cada conjunto de elementos que hemos considerado significativos: guías, usuarios, rutas, tramos, incidencias y notificaciones.



Las tablas tienen un nombre identificativo en singular de lo que se puede encontrar en cada fila de ésta.

Para relacionar unas tablas con otras hemos creado “tablas nexa” cuyo contenido no es otro que los identificadores de los elementos que agrupan. Las tablas son: usuarioguía, rutausuario, tramoruta, incidenciusuario.

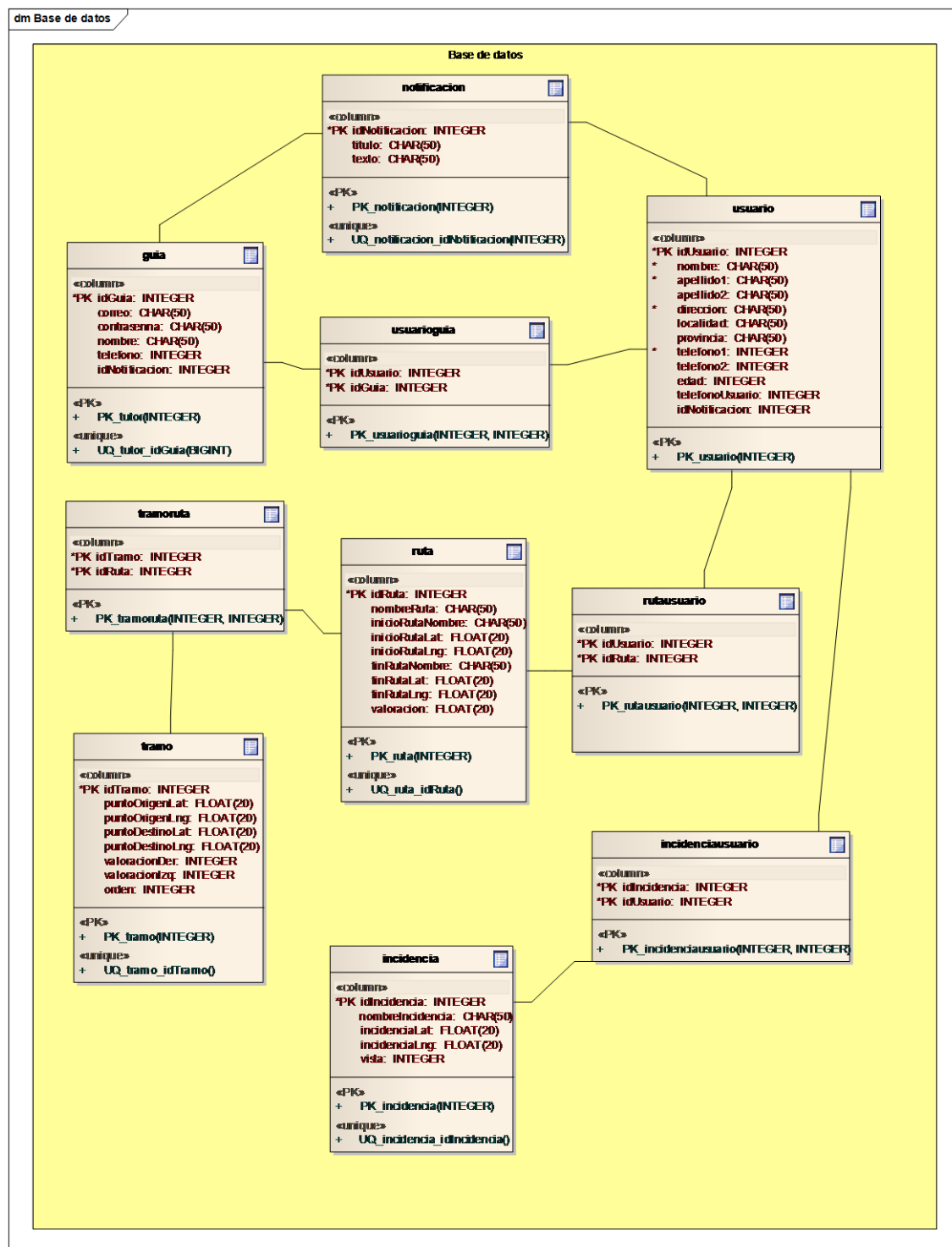


Ilustración 12: Diagrama de la base de datos

### 4.3. API

Del inglés *Application Programming Interface* (Interfaz de programación de aplicaciones), es el conjunto de funciones y procedimientos que establecen comunicación con la base de datos. Nos ofrece sencillez y transparencia a la hora de seleccionar, modificar o eliminar datos desde cada una de las aplicaciones.

En este caso, se trata de 47 archivos de extensión “.php” contruidos mediante PHP orientado a objetos y donde el fichero en el que se encuentra toda la lógica interna y consultas directas a la base de datos es *funciones\_db.php*, el que realiza la conexión con esta es *connectbd.php* y donde se configura la base de datos a la que va a conectarse y los datos de usuario y contraseña para ello es *config.php*.

Los archivos PHP para la web son los siguientes:

- *addGuia.php*: inserta en la base de datos el nuevo guía registrado en el sistema.
- *addRuta.php*: dado un usuario y toda la información de una ruta (nombre de la ruta, nombre y coordenadas de la calle de inicio y nombre y coordenadas de la calle de final de la ruta), añade a la base de datos una nueva ruta asociándola al usuario.
- *addTramo.php*: dada una ruta con todas las coordenadas de los tramos y sus valoraciones, asocia a la ruta cada tramo introducido, añade a la base de datos los tramos con sus valoraciones y actualiza la valoración de la ruta.
- *addUsuario.php*: dado un guía y la información del usuario (Nombre, Apellidos, Dirección (Calle, Localidad, Provincia), Teléfonos y Edad), asocia el guía y el usuario en la base de datos y añade al nuevo usuario en la base de datos.
- *cerrarsesion.php*: se encarga de cerrar la sesión de guía previamente había iniciado sesión en el sistema.
- *changeNombreRuta.php*: dado la ruta y el nombre de la ruta, modifica en la base de datos el nombre de la ruta.
- *changeTramosRuta.php*: dado una ruta y unos tramos, modifica en la base de datos la valoración de los tramos modificados de la ruta y de la propia ruta.
- *changeUsuario.php*: dado un usuario, el nombre, los apellidos, la dirección (Calle, Localidad y Provincia), los teléfonos y la edad del usuario actualiza la información del usuario en la base de datos.
- *deleteIdRuta.php*: dada una ruta, elimina esa ruta con los tramos que tenga asociados.
- *deleteIdUsuario.php*: dado el usuario, elimina a dicho usuario de la base de datos junto a todas sus rutas asociadas.
- *getCoordenadasRuta.php*: dado una ruta, consulta la base de datos y nos devuelve las coordenadas de inicio y final de la ruta.
- *getUsuariosGuia.php*: contiene una función que consulta la base de datos y nos devuelve todos los usuarios que tiene asociado un guía.
- *getRutasGuia.php*: contiene una función que consulta la base de datos y nos devuelve todas las rutas que tiene asociado todos los usuarios de un guía.
- *getRutasUsuario.php*: contiene una función que consulta la base de datos y nos devuelve todas las rutas que tiene asociado un usuario.

- *getTramosRuta.php*: contiene una función que consulta la base de datos y nos devuelve todos los tramos que tiene asociada una ruta.
- *getUsuario.php*: dado un usuario, consulta la base de datos y nos devuelve toda la información de dicho usuario.
- *importar.php*: dado un usuario y un guía, actualiza la base de datos cambiando al usuario de guía.
- *login.php*: consulta en la base de datos si el guía que está accediendo al sistema ya está registrado en el sistema.

Los archivos PHP para la app del usuario son los siguientes:

- *addGuiaApp.php*: inserta en la base de datos el nuevo guía registrado en el sistema.
- *addIncidenciaApp.php*: dado un usuario y toda la información de una incidencia (nombre y coordenadas de la incidencia), añade a la base de datos una nueva incidencia asociándola al usuario.
- *addNotificacionIncidenciaAGuiaApp.php*: dado un usuario y una incidencia, asocia la incidencia con el guía del usuario.
- *getGuia.php*: dado un guía, nos devuelve toda la información de este guía (correo, contraseña, nombre y teléfono).
- *getNotificacionesUsuarioApp.php*: dado un usuario, nos devuelve todas las notificaciones de este.
- *getRutasUsuarioApp.php*: dado un usuario, devuelve todas las rutas que tiene asociadas este.
- *getTramosRutaApp.php*: dado una ruta, devuelve todos los tramos que tiene asociadas dicha ruta.
- *getUsuario.php*: dado un usuario, consulta la base de datos y nos devuelve toda la información de dicho usuario.
- *getUsuariosGuiaApp.php*: dado un guía, devuelve todos los usuarios de dicho guía.
- *loginApp.php*: consulta en la base de datos si el guía que está accediendo al sistema ya está registrado.

Los archivos PHP para la app del tutor son los siguientes:

- *addGuiaApp.php*: inserta en la base de datos el nuevo guía registrado en el sistema.
- *deleteNotificacionGuiaApp.php*: dado un guía elimina las notificaciones del mismo.
- *getGuia.php*: dado un guía nos devuelve toda la información de este guía (correo, contraseña, nombre y teléfono).
- *getIncidenciasGuiaApp.php*: dado un guía, nos devuelve todas las incidencias de los usuarios del mismo.
- *getNotificacionesGuiaApp.php*: dado un guía, nos devuelve todas las notificaciones de este.

- *getRutasGuiaApp.php*: dado un guía, nos devuelve todas las rutas de los usuarios del mismo.
- *getTramosRutaApp.php*: dado una ruta, que devuelve todos los tramos que tiene asociadas dicha ruta.
- *getUsuariosGuiaApp.php*: dado un guía, nos devuelve todos los usuarios de este.
- *getUsuario.php*: dado un usuario, consulta la base de datos y nos devuelve toda la información de dicho usuario.
- *loginApp.php*: consulta en la base de datos si el guía que está accediendo al sistema ya está registrado.
- *setVistaIncidenciaApp.php*: dada una incidencia, modifica la vista de la misma.

Además de estos, contamos con 13 archivos de extensión “.html” cuyo objetivo es probar el correcto funcionamiento de los anteriores, sin tener que ejecutar para ello una de las aplicaciones, y poder de este modo tener un punto intermedio entre la base de datos y las aplicaciones en sí que faciliten el mantenimiento.

## 4.4. Arquitectura de la web (WebTutor)

El propósito de esa sección es explicar las diferentes funcionalidades de la aplicación web a través de casos de uso y diagramas de actividades. Además de analizar las diferentes clases utilizadas en esta aplicación ayudándonos de una breve explicación para cada una de ellas.

### Casos de uso



Ilustración 13: Diagrama general de casos de uso de la página Web

## AÑADIR RUTA

Tabla 1: Caso de uso: Añadir ruta - Aplicación WebTutor		
Descripción	El guía añade una nueva ruta seleccionando el punto de inicio y final de la ruta.	
Restricciones	Pre-Condiciones	El guía debe haber iniciado sesión en el sistema, por medio del inicio de sesión o registro y haber pulsado el botón “Añadir nueva ruta”.
	Post-Condiciones	En la página principal del guía se añadirá la nueva ruta sin valorar.
Secuencia de eventos	<p><b>P.1.</b> El guía pulsa el botón “Añadir nueva ruta”.</p> <p><b>P.2.</b> El sistema muestra una ventana con los datos a rellenar por el guía para añadir la nueva ruta.</p> <p><b>P.3.</b> El guía pulsa el botón “Usuarios”.</p> <p><b>P.4.</b> El sistema muestra los usuarios del guía que inicio la sesión.</p> <p><b>P.5.</b> El guía selecciona el usuario al que se va añadir la nueva ruta.</p> <p><b>P.6.</b> El sistema muestra un espacio para introducir el nombre de la ruta.</p> <p><b>P.7.</b> El guía introduce el nombre de la ruta.</p> <p><b>P.8.</b> El guía pulsa el botón “Aceptar”.</p> <p><b>P.9.</b> El sistema muestra un espacio para introducir el inicio de la ruta.</p> <p><b>P.10.</b> El guía selecciona el tipo de vía del inicio de la ruta.</p> <p><b>P.11.</b> El guía introduce el inicio de la ruta.</p> <p><b>P.12.</b> El guía pulsa el botón “Comprobar”.</p> <p><b>P.13.</b> El sistema muestra el mapa con el marker del inicio de la ruta.</p> <p><b>P.13.1.</b> El guía pulsa el botón “Ups, me equivoque”.</p> <p><b>P.13.1.1.</b> El sistema nos lleva a P.9.</p> <p><b>P.13.2.</b> El guía pulsa el botón “Modificar puntero”.</p> <p><b>P.13.2.1.</b> El sistema hace editable el marker de inicio de la ruta.</p> <p><b>P.13.2.2.</b> El guía selecciona el marker y los desplaza hasta el lugar correcto del mapa.</p> <p><b>P.13.2.3.</b> El guía pulsa el botón “Ok, inicio modificado”.</p> <p><b>P.13.2.4.</b> El sistema nos lleva a P.13.</p> <p><b>P.13.3.</b> El guía pulsa el botón “Correcto”.</p> <p><b>P.13.3.1.</b> El sistema nos muestra un espacio para introducir el final de la ruta.</p> <p><b>P.13.3.2.</b> El guía selecciona el tipo de vía del final de la ruta.</p> <p><b>P.13.3.3.</b> El guía introduce el final de la ruta.</p> <p><b>P.13.3.4.</b> El guía pulsa el botón “Comprobar”.</p> <p><b>P.13.3.5.</b> El sistema muestra el mapa con el marker del final de la ruta.</p> <p><b>P.13.3.5.1.</b> El guía pulsa el botón “Ups, me equivoque”.</p> <p><b>P.13.3.5.1.1.</b> El sistema nos lleva a P.13.3.1.</p> <p><b>P.13.3.5.2.</b> El guía pulsa el botón “Modificar puntero”.</p> <p><b>P.13.3.5.2.1.</b> El sistema hace editable el marker de inicio de la ruta.</p> <p><b>P.13.3.5.2.2.</b> El guía selecciona el marker y los desplaza hasta el lugar correcto del mapa.</p> <p><b>P.13.3.5.2.3.</b> El guía pulsa el botón “Ok, final modificado”.</p>	

	<p><b>P.13.3.5.2.4.</b> El sistema nos lleva a P.13.3.5.</p> <p><b>P.13.3.5.3.</b> El guía pulsa el botón “Correcto”.</p> <p><b>P.13.3.5.3.1.</b> El guía pulsa el botón “Aceptar”.</p> <p><b>P.13.3.5.3.1.1.</b> El sistema añade a la base de datos la nueva ruta asociándola al usuario previamente seleccionado.</p> <p><b>P.13.3.5.3.1.2.</b> El sistema nos lleva a la página principal al menú de rutas.</p> <p><b>P.13.3.5.3.2.</b> El guía pulsa el botón “Cancelar”.</p> <p><b>P.13.3.5.3.2.1.</b> El sistema nos lleva a la página principal al menú de rutas.</p>
--	---

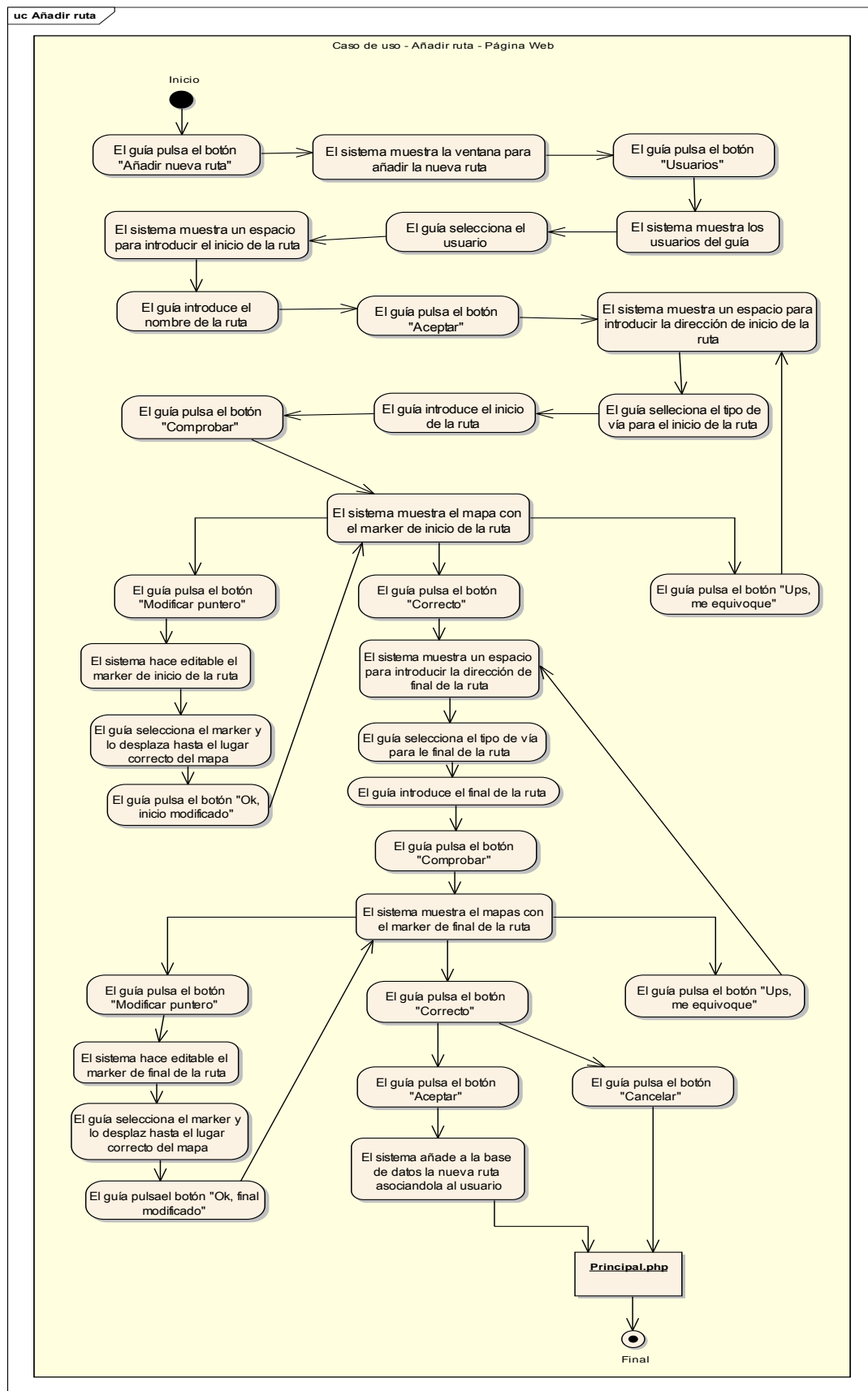


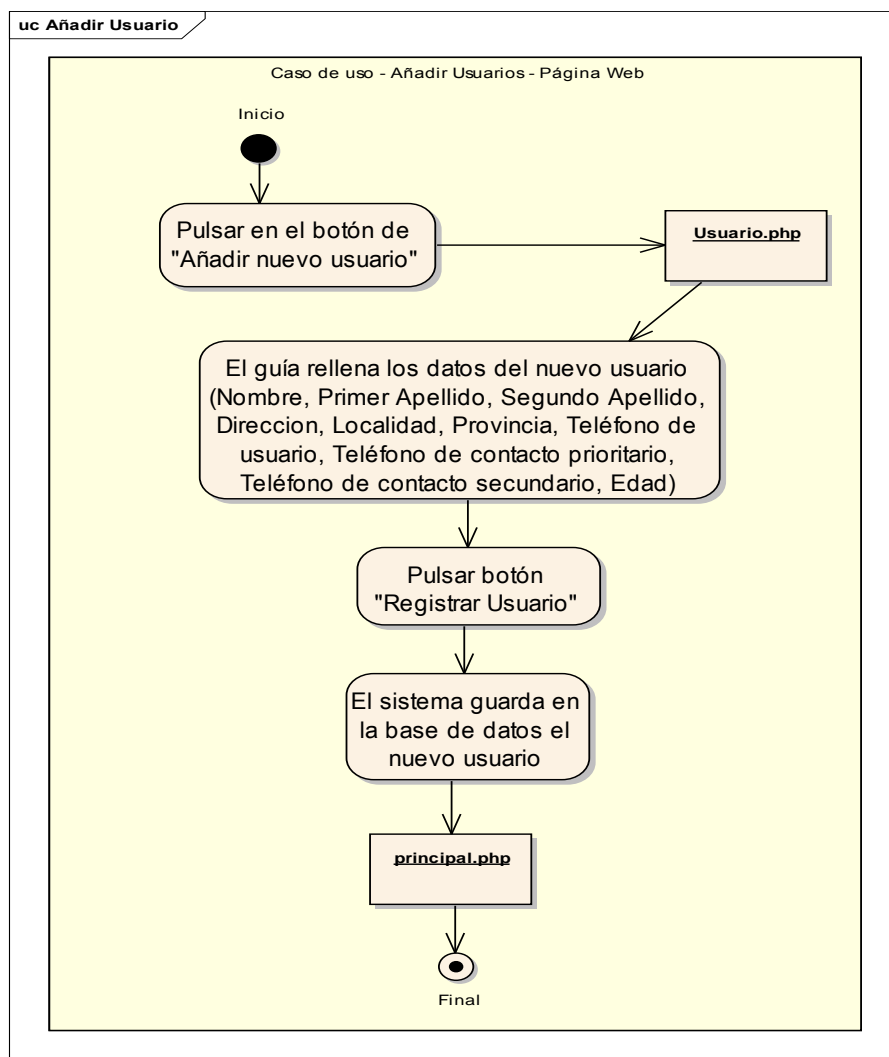
Ilustración 14: Diagrama de actividad del caso de uso Añadir Ruta de la página Web



## AÑADIR USUARIO

**Tabla 2: Caso de uso: Añadir Usuario - Aplicación WebTutor**

Descripción	Añadir nuevos usuarios asociados al guía que inicio la sesión.	
Restricciones	Pre-Condiciones	Haber iniciado sesión en el sistema, estar en el menú usuarios y haber pulsado el botón de “Añadir nuevo usuario”.
	Post-Condiciones	Añadir nuevos usuarios.
Secuencia de eventos	<p><b>P.1.</b> Pulsar botón “Añadir nuevo usuario”.</p> <p><b>P.2.</b> El sistema nos lleva a la página usuario.php</p> <p><b>P.3.</b> El guía completa los campos necesarios para el registro del nuevo usuario (Nombre, Primer Apellido, Segundo Apellido, Dirección, Localidad, Provincia, Teléfono del usuario, Teléfono de contacto prioritario, Teléfono de contacto secundario, Edad)</p> <p><b>P.4.</b> El guía pulsa el botón “Registrar usuario”</p> <p><b>P.5.</b> El sistema guarda en la base de datos el nuevo usuario asociado al guía.</p> <p><b>P.6.</b> El sistema nos lleva a la página principal en menú de usuarios.</p>	



*Ilustración 15: Diagrama de actividad del caso de uso Añadir Usuario de la página Web*

## CERRAR SESIÓN

Tabla 3: Caso de uso: Cerrar Sesión - Aplicación WebTutor		
Descripción	Cerrar la sesión del guía logueado.	
Restricciones	Pre-Condiciones	Haber iniciado sesión en el sistema.
	Post-Condiciones	Poder registrar otro guía o loguearse nuevamente.
Secuencia de eventos	<p><b>P.1.</b> Pulsar botón “Cerrar sesión”.</p> <p><b>P.2.</b> El sistema cierra la sesión del guía que inicio sesión en el sistema.</p> <p><b>P.3.</b> El sistema muestra un mensaje indicando que la sesión se ha cerrado correctamente.</p> <p><b>P.4.</b> El sistema nos lleva a la página index.php</p>	

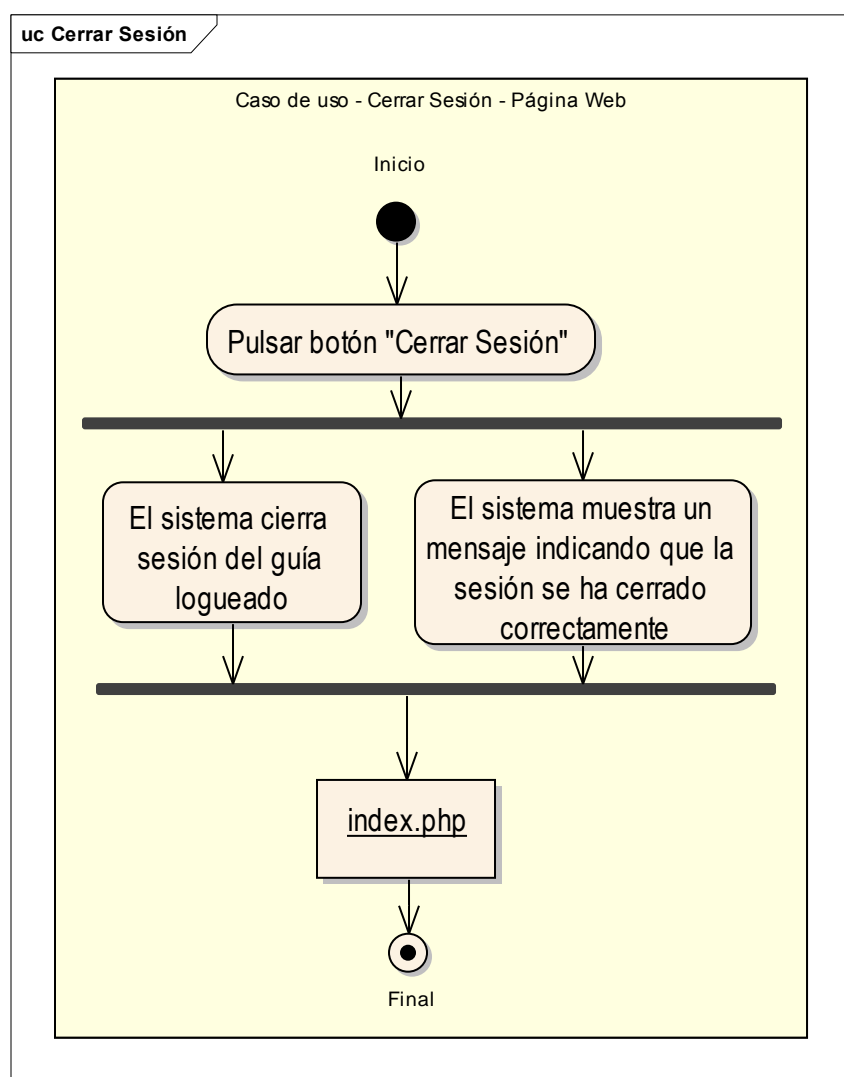
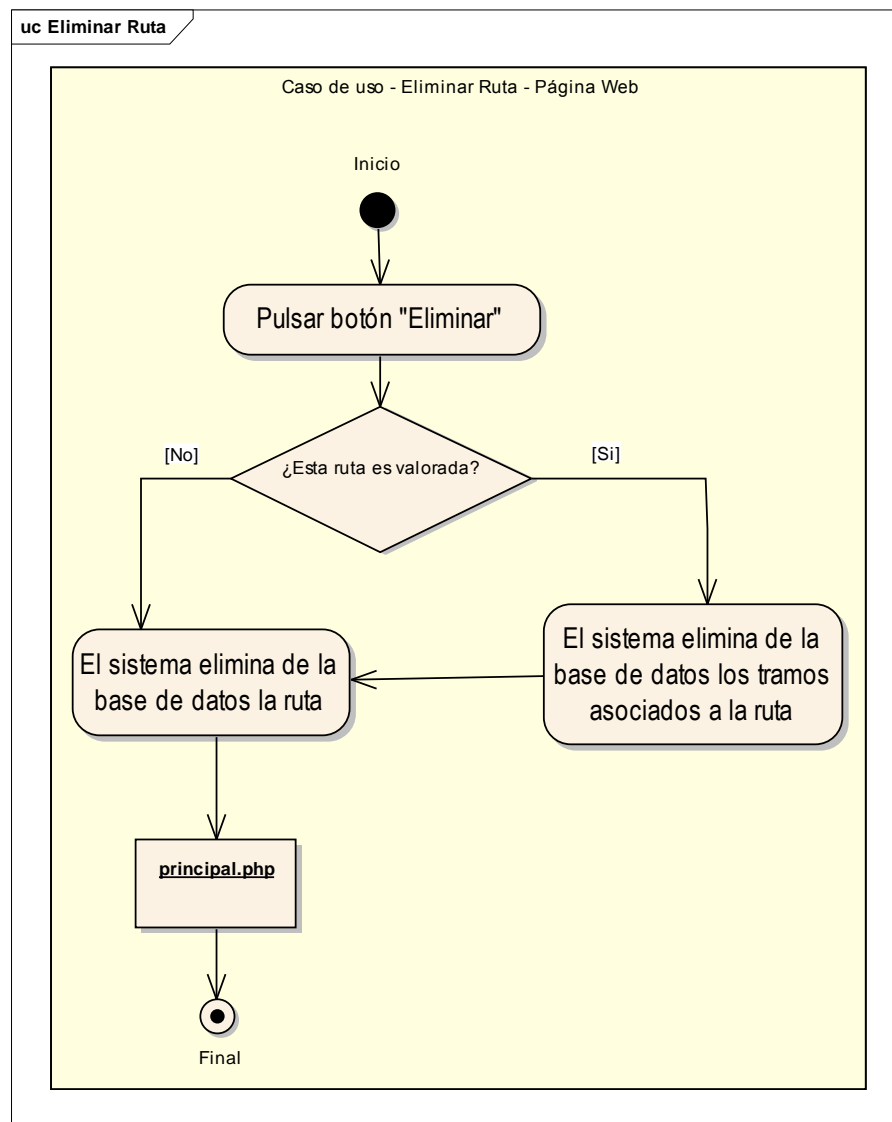


Ilustración 16: Diagrama de actividad del caso de uso Cerrar sesión de la página Web

## ELIMINAR RUTA

**Tabla 4: Caso de uso: Eliminar Ruta - Aplicación WebTutor**

Descripción	Eliminar Rutas de un usuario.	
Restricciones	Pre-Condiciones	Iniciar sesión en el sistema, haber pulsado rutas en el menú y pulsar eliminar en la ruta deseada.
	Post-Condiciones	El guía elimina la ruta seleccionada.
Secuencia de eventos	<p><b>P.1.</b> Pulsar eliminar en una ruta.</p> <p><b>P.2.</b> El sistema busca en la bases de datos si la ruta está valorada.</p> <p><b>P.2.1.</b> La ruta seleccionada está valorada.</p> <p><b>P.2.1.1.</b> El sistema elimina de la base de datos los tramos asociados a la ruta seleccionada.</p> <p><b>P.2.1.2.</b> El sistema elimina de la base de datos la ruta seleccionada.</p> <p><b>P.2.2.</b> La ruta seleccionada no está valorada.</p> <p><b>P.2.2.1.</b> El sistema elimina de la base de datos la ruta seleccionada.</p> <p><b>P.3.</b> El sistema nos lleva a la página principal en el menú de rutas</p>	



*Ilustración 17: Diagrama de actividad del caso de uso Eliminar Ruta de la página Web*

## ELIMINAR USUARIO

Tabla 5: Caso de uso: Eliminar Usuario - Aplicación WebTutor		
Descripción	Poder eliminar usuarios.	
Restricciones	Pre-Condiciones	Haber iniciado sesión en el sistema, estar en el menú usuarios y tener al menos un usuario.
	Post-Condiciones	Eliminar el usuario deseado.
Secuencia de eventos	<p><b>P.1.</b> Pulsar botón “Eliminar”.</p> <p><b>P.2.</b> El sistema nos muestra los usuarios del guía que ha iniciado sesión.</p> <p><b>P.3.</b> El guía selecciona el usuario que quiere eliminar.</p> <p><b>P.4.</b> El sistema busca en la base de datos si el usuario seleccionado tiene rutas asociadas.</p> <p><b>P.4.1.</b> El usuario si tiene rutas asociadas.</p> <p><b>P.4.1.1.</b> El sistema elimina de la bases de datos los tramos de las rutas.</p> <p><b>P.4.1.2.</b> El sistema elimina de la base de datos las rutas.</p> <p><b>P.4.1.3.</b> El sistema elimina de la bases de datos al usuario seleccionado.</p> <p><b>P.4.2.</b> El usuario no tiene rutas asociadas.</p> <p><b>P.4.2.1.</b> El sistema elimina de la base de datos al usuario seleccionado.</p> <p><b>P.5.</b> El sistema nos lleva a la página principal en el menú usuarios.</p>	

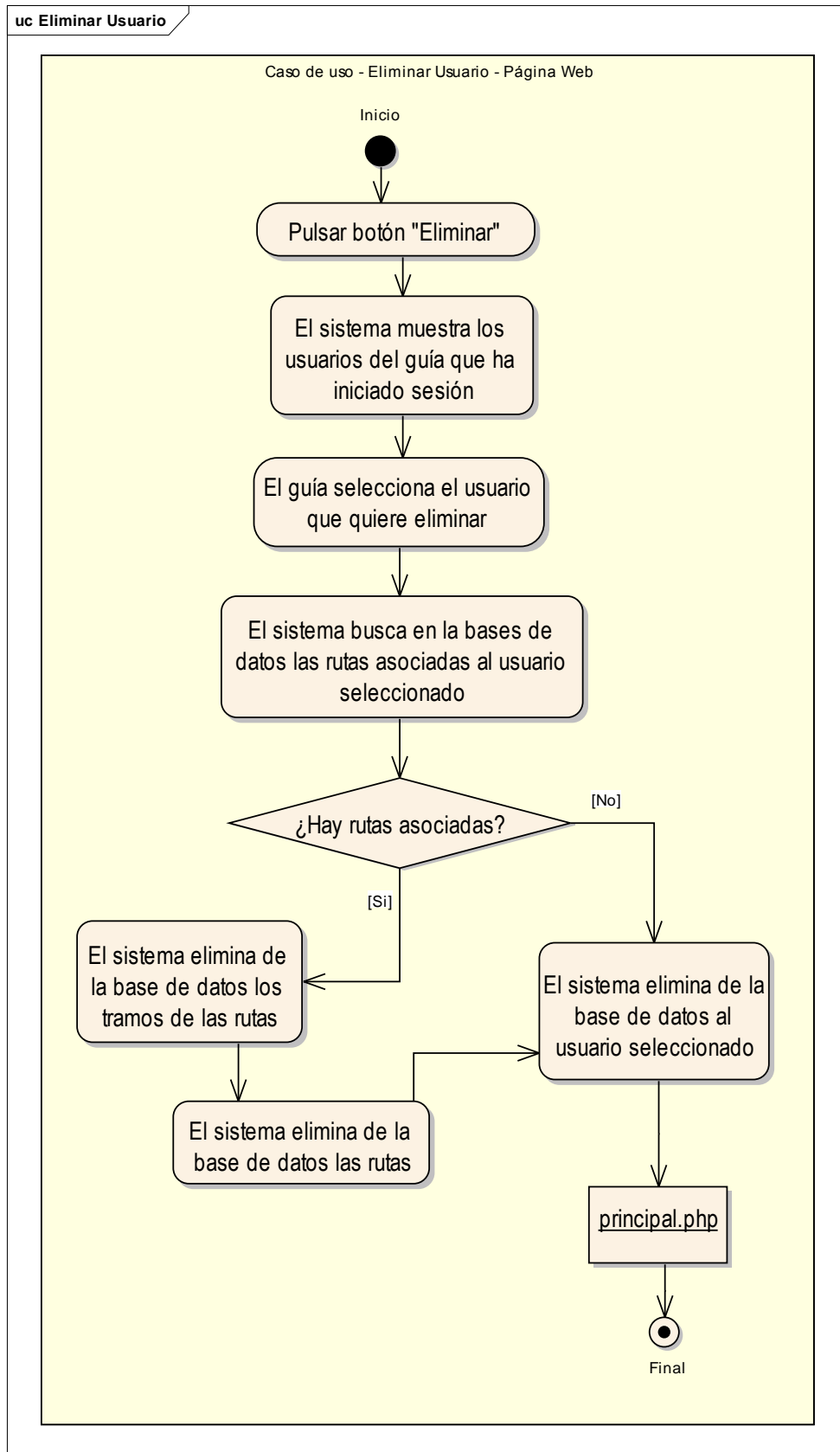


Ilustración 18: Diagrama de actividad del caso de uso Eliminar Usuario de la página Web

## EXPORTAR USUARIOS

Tabla 6: Caso de uso: Exportar Usuarios - Aplicación WebTutor		
Descripción	Realizar un archivo con usuarios para mandárselo a otros guías.	
Restricciones	Pre-Condiciones	Iniciar sesión en el sistema y haber pulsado configuración en el menú.
	Post-Condiciones	El guía podrá enviar usuarios de otros guías.
Secuencia de eventos	<b>P.1.</b> El guía selecciona los usuarios que quiera enviar a otros guías. <b>P.2.</b> El guía Pulsa el botón “Aceptar”. <b>P.3.</b> El sistema genera un archivo con los usuarios que el guía haya seleccionado.	

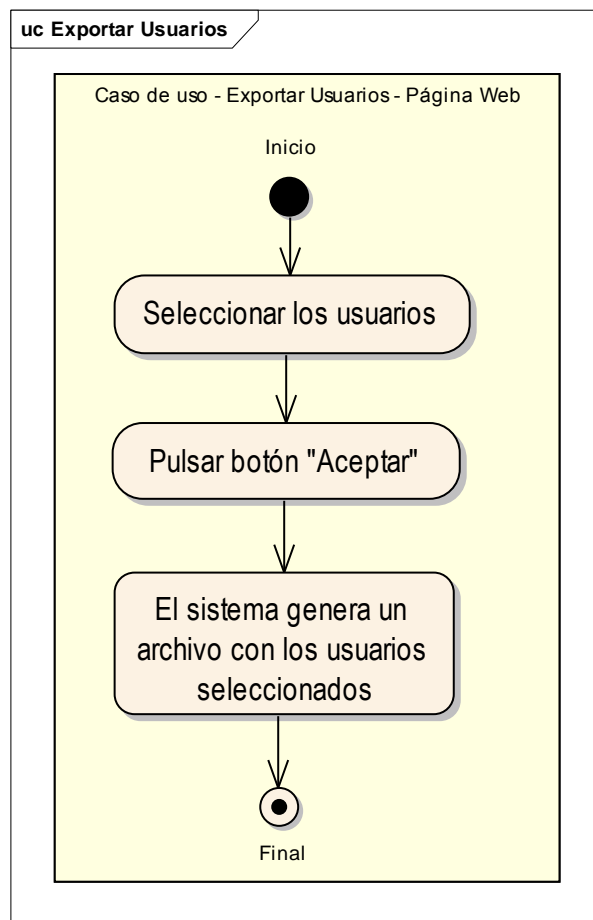


Ilustración 19: Diagrama de actividad del caso de uso Exportar usuarios de la página Web

## IMPORTAR USUARIOS

Tabla 7: Caso de uso: Importar Usuarios - Aplicación WebTutor		
Descripción	Añadir usuarios de otros guías a sus propios usuarios.	
Restricciones	Pre-Condiciones	Iniciar sesión en el sistema, haber pulsado configuración en el menú y haber recibido el archivo con los usuarios de otro guía.
	Post-Condiciones	El guía podrá añadir usuarios de otros guías.
Secuencia de eventos	<p><b>P.1.</b> Pulsar el botón “Seleccionar archivo”.</p> <p><b>P.2.</b> El usuario selecciona el archivo que previamente recibió de otro usuario.</p> <p><b>P.3.</b> Pulsar botón “Aceptar”.</p> <p><b>P.4.</b> El sistema actualiza la base de datos, cambiando los usuarios de guía.</p>	

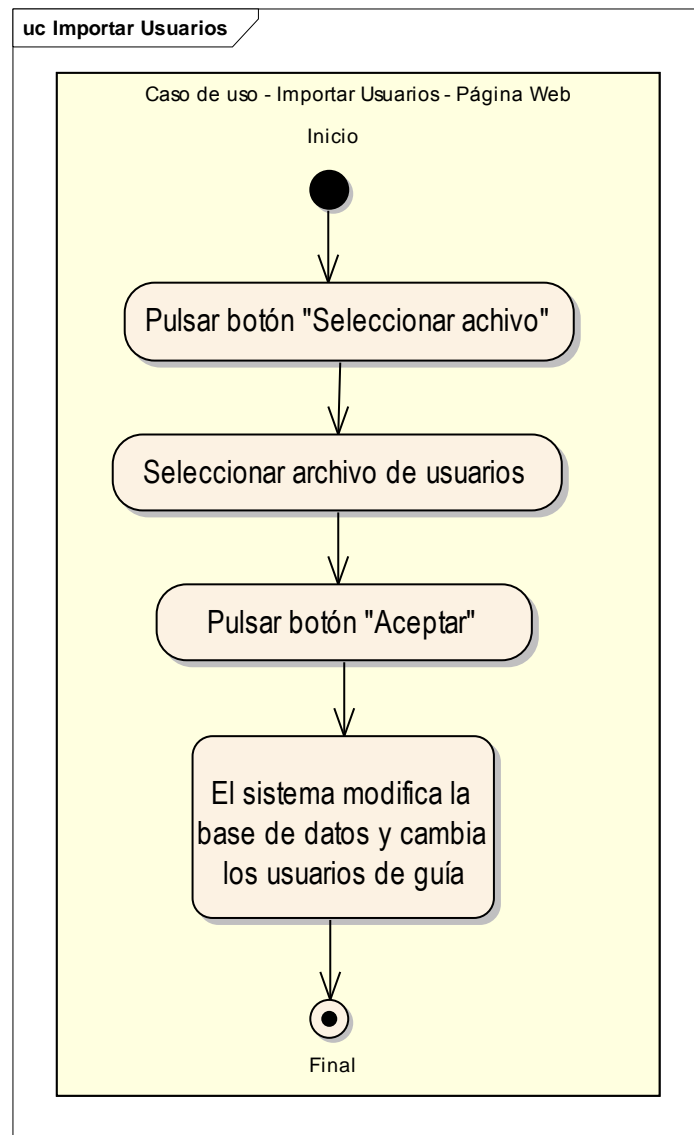


Ilustración 20: Diagrama de actividad del caso de uso Importar Usuarios de la página Web

## INICIO SESIÓN

Tabla 8: Caso de uso: Inicio Sesión - Aplicación WebTutor		
Descripción	Inicio de la sesión de un guía en el sistema.	
Restricciones	Pre-Condiciones	Estar registrado en el sistema.
	Post-Condiciones	El guía inicia sesión en el sistema.
Secuencia de eventos	<p><b>P.1.</b> Introducir los datos requeridos para el inicio de sesión (Correo y Contraseña).</p> <p><b>P.2.</b> Pulsar botón “Entrar”</p> <p><b>P.2.1.</b> El sistema busca en la base de datos.</p> <p><b>P.2.1.1.</b> El sistema encuentra un guía que coincide con ese correo y contraseña.</p> <p><b>P.2.1.1.1.</b> El sistema nos lleva a la página principal</p> <p><b>P.2.1.2.</b> El sistema no encuentra un guía que coincida con ese correo y contraseña.</p> <p><b>P.2.1.2.1.</b> El sistema muestra un mensaje de error, indicando que el correo o la contraseña no son correctos.</p> <p><b>P.2.1.2.2.</b> El sistema nos lleva a la página index</p>	

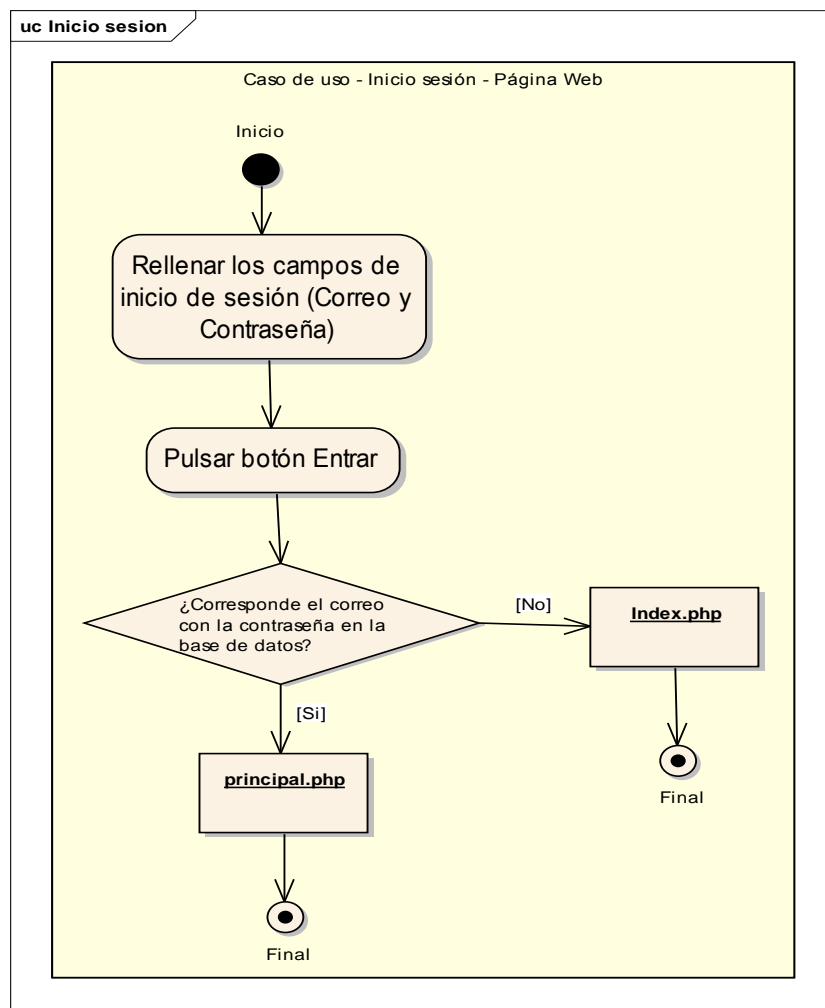


Ilustración 21: Diagrama de actividad del caso de uso Inicio sesión de la página Web



## MENÚ CONFIGURACIÓN

Tabla 9: Caso de uso: Menú Configuración - Aplicación WebTutor		
Descripción	Mostrar el menú configuración de un guía.	
Restricciones	Pre-Condiciones	Haber iniciado sesión en el sistema y haber pulsado configuración el menú del guía.
	Post-Condiciones	El guía podrá importar o exportar sus usuarios.
Secuencia de eventos	<b>P.1.</b> Pulsar configuración del menú del guía. <b>P.2.</b> El sistema muestra las opciones de importar y exportar sus usuarios.	

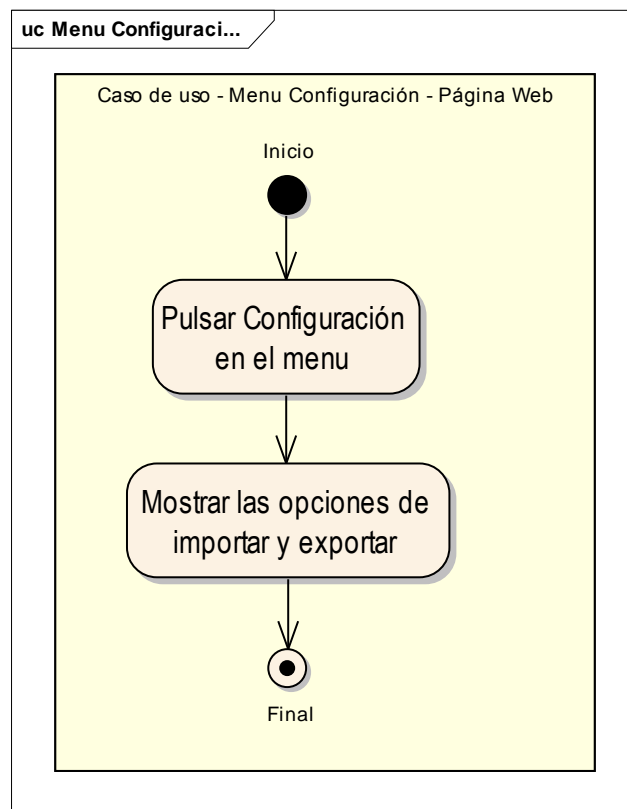
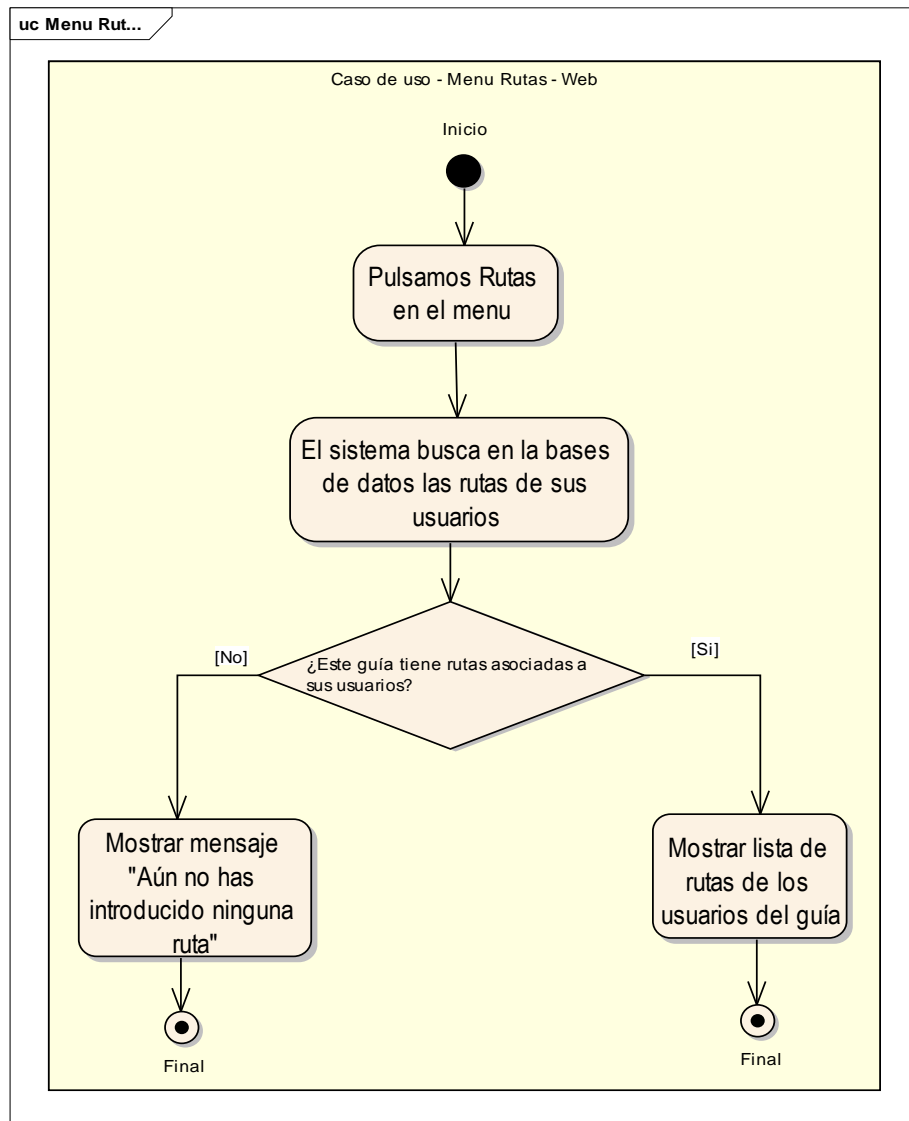


Ilustración 22: Diagrama de actividad del caso de uso Menú configuración de la página Web

## MENÚ RUTAS

**Tabla 10: Caso de uso: Menú Rutas - Aplicación WebTutor**

Descripción	Mostrar el menú de Rutas de los usuarios de un guía.	
Restricciones	Pre-Condiciones	Haber iniciado sesión en el sistema y haber pulsado rutas en el menú.
	Post-Condiciones	El guía podrá visualizar todas las rutas de sus usuarios registrados y registrar nuevas rutas.
Secuencia de eventos	<p><b>P.1.</b> Pulsar rutas del menú del guía.</p> <p><b>P.2.</b> El sistema busca en la bases de datos las rutas de los usuarios del guía que ha iniciado sesión.</p> <p><b>P.2.1.</b> El sistema encuentra rutas de los usuarios del guía.</p> <p><b>P.2.1.1.</b> El sistema muestra las rutas de los usuarios del guía.</p> <p><b>P.2.2.</b> EL sistema no encuentra rutas de los usuarios del guía.</p> <p><b>P.2.2.1.</b> El sistema muestra el mensaje “Aún no has introducido ninguna ruta”.</p>	

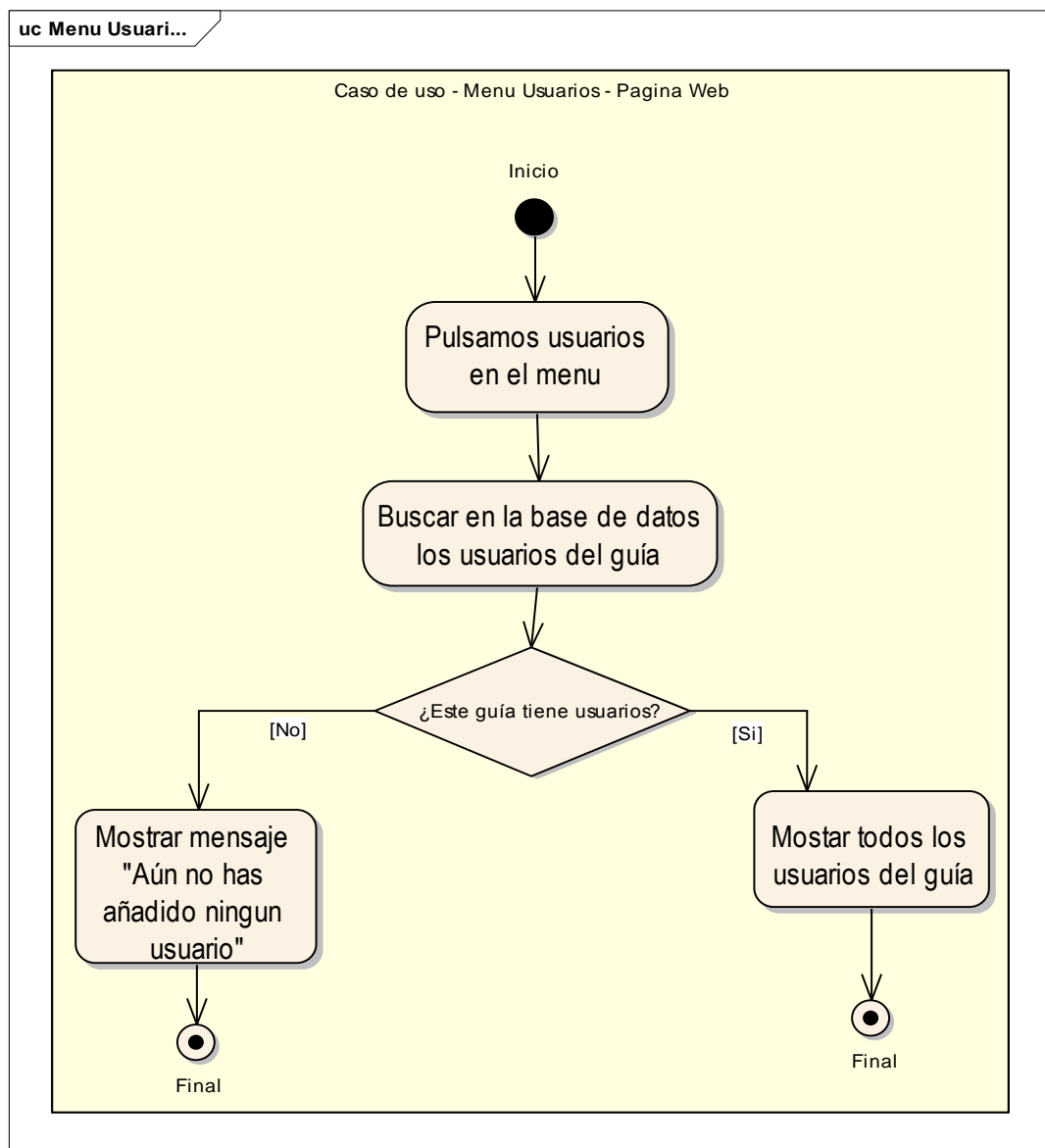


*Ilustración 23: Diagrama de actividad del caso de uso Menú rutas de la página Web*

## MENÚ USUARIOS

**Tabla 11: Caso de uso: Menú usuarios - Aplicación WebTutor**

Descripción	Mostrar el menú de usuarios de un guía.	
Restricciones	Pre-Condiciones	Haber iniciado sesión en el sistema.
	Post-Condiciones	El guía podrá visualizar todos sus usuarios registrados y registrar nuevos usuarios.
Secuencia de eventos	<p><b>P.1.</b> Pulsar usuarios del menú del guía.</p> <p><b>P.2.</b> El sistema busca en la bases de datos los usuarios del guía que ha iniciado sesión.</p> <p><b>P.2.1.</b> El sistema encuentra usuarios del guía.</p> <p><b>P.2.1.1.</b> El sistema muestra los usuarios del guía.</p> <p><b>P.2.2.</b> EL sistema no encuentra usuarios del guía.</p> <p><b>P.2.2.1.</b> El sistema muestra el mensaje “Aún no has añadido ningún usuario”.</p>	



*Ilustración 24: Diagrama de actividad del caso de uso Menú usuarios de la página Web*

## MODIFICAR RUTA

Tabla 12: Caso de uso: Modificar ruta - Aplicación WebTutor		
Descripción	Poder modificar rutas que ya están en el sistema.	
Restricciones	Pre-Condiciones	Haber iniciado sesión en el sistema, estar en el menú rutas y tener al menos una ruta.
	Post-Condiciones	Modificar el nombre y tramos de la ruta deseada.
Secuencia de eventos	<p><b>P.1.</b> Pulsar botón “Modificar”.</p> <p><b>P.2.</b> El sistema nos lleva a la página de modificar ruta.</p> <p><b>P.2.1.</b> El guía pulsa el botón “modificar Nombre”.</p> <p><b>P.2.1.1.</b> El sistema muestra una ventana con el nombre de la ruta actual para poder modificarlo.</p> <p><b>P.2.1.2.</b> El guía modifica el nombre de la ruta.</p> <p><b>P.2.1.2.1.</b> El guía pulsa el botón “Modificar”.</p> <p><b>P.2.1.2.1.1.</b> El sistema modifica en la base de datos el nombre de la ruta, sustituyéndolo por el introducido por el guía.</p> <p><b>P.2.1.2.1.2.</b> El sistema nos lleva a la página modificar ruta cerrando la ventana.</p> <p><b>P.2.1.2.2.</b> El guía pulsa el botón “Cancelar”.</p> <p><b>P.2.1.2.2.1.</b> El sistema nos lleva a la página modificar ruta cerrando la ventana.</p> <p><b>P.2.2.</b> El guía pulsa el botón “modificar Tramos”.</p> <p><b>P.2.2.1.</b> El sistema muestra los tramos de la ruta.</p> <p><b>P.2.2.1.1.</b> El guía pulsa el botón “Modificar Tramo X”.</p> <p><b>P.2.2.1.1.1.</b> El sistema muestra el sistema de valoración de aceras (Izquierda y derecha) y el botón de modificar.</p> <p><b>P.2.2.1.1.2.</b> El guía valorará los factores mostrados mediante los iconos, para cada acera.</p> <p><b>P.2.2.1.1.3.</b> El guía pulsará el botón “Modificar”</p> <p><b>P.2.2.1.2.</b> El guía pulsa el botón “Volver”.</p> <p><b>P.2.2.1.2.1.</b> El sistema nos lleva a la página de modificar ruta.</p> <p><b>P.2.3.</b> El guía pulsa el botón “Finalizar”.</p> <p><b>P.2.3.1.</b> El sistema nos lleva a la página principal en el menú rutas.</p>	

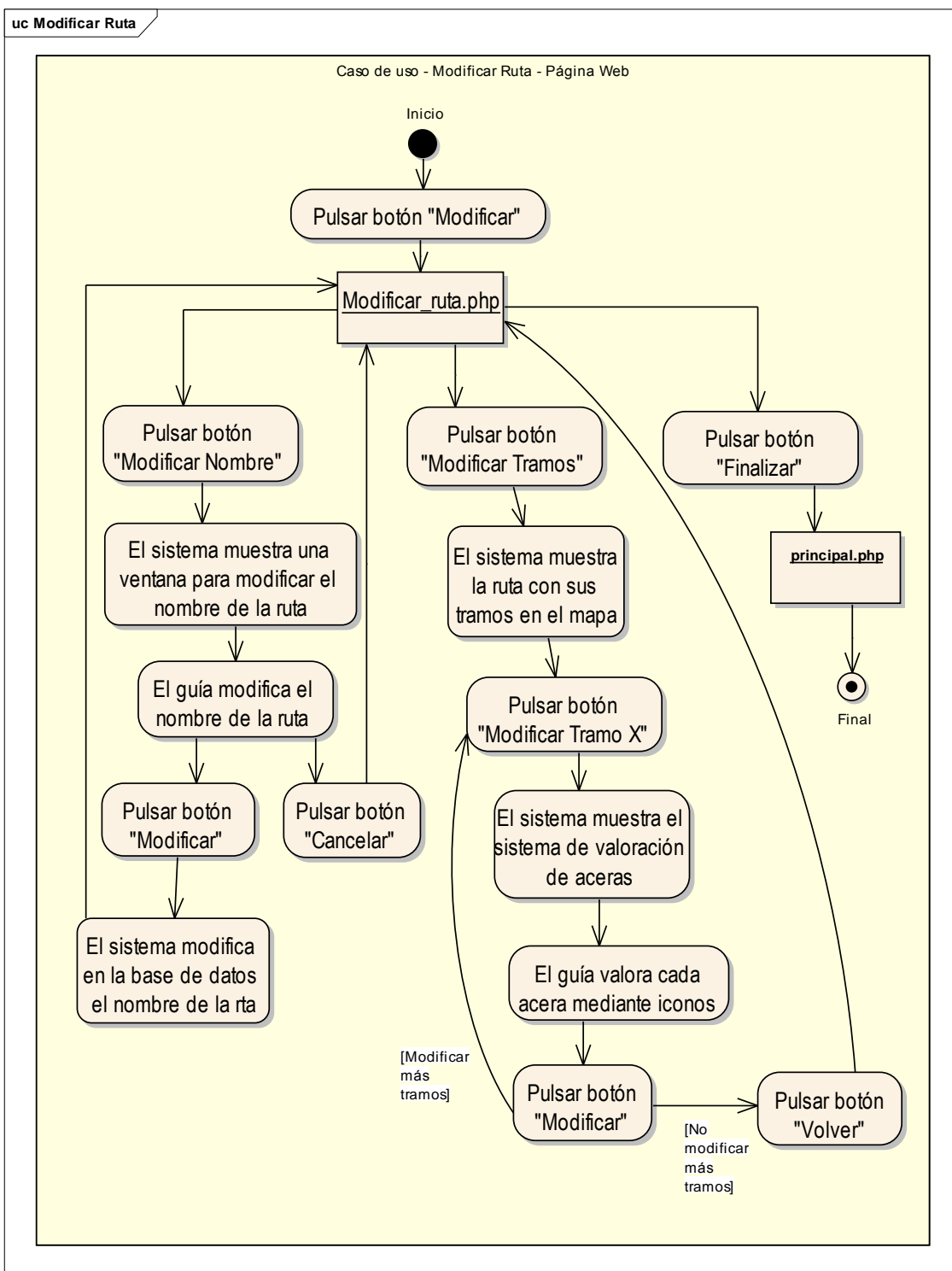


Ilustración 25: Diagrama de actividad del caso de uso Modificar Ruta de la página Web

## MODIFICAR USUARIO

Tabla 13: Caso de uso: Modificar Usuario - Aplicación WebTutor		
Descripción	Poder modificar usuarios que ya están en el sistema.	
Restricciones	Pre-Condiciones	Haber iniciado sesión en el sistema, estar en el menú usuarios y tener al menos un usuario.
	Post-Condiciones	Modificar la información del usuario deseado.
Secuencia de eventos	<p><b>P.1.</b> Pulsar botón “Modificar”.</p> <p><b>P.2.</b> El sistema nos muestra los usuarios del guía que ha iniciado sesión.</p> <p><b>P.3.</b> El guía selecciona el usuario que quiere modificar.</p> <p><b>P.4.</b> El sistema nos lleva a <i>modificar_usuario.php</i>.</p> <p><b>P.5.</b> El guía modifica los datos del usuario seleccionado.</p> <p><b>P.5.1.</b> El usuario modifica el nombre.</p> <p><b>P.5.2.</b> El usuario modifica el primer apellido.</p> <p><b>P.5.3.</b> El usuario modifica el segundo apellido.</p> <p><b>P.5.4.</b> El usuario modifica la dirección.</p> <p><b>P.5.5.</b> El usuario modifica la localidad.</p> <p><b>P.5.6.</b> El usuario modifica la provincia.</p> <p><b>P.5.7.</b> El usuario modifica el teléfono del usuario.</p> <p><b>P.5.8.</b> El usuario modifica el teléfono de contacto prioritario.</p> <p><b>P.5.9.</b> El usuario modifica el teléfono de contacto secundario.</p> <p><b>P.5.10.</b> El usuario modifica la edad.</p> <p><b>P.6.</b> El usuario pulsa el botón “Guardar Cambios”.</p> <p><b>P.7.</b> El sistema modifica en la base de datos los datos modificados por el guía.</p> <p><b>P.8.</b> El sistema nos muestra un mensaje indicando que el usuario ha sido modificado correctamente.</p> <p><b>P.9.</b> El guía pulsa el botón “Aceptar”.</p> <p><b>P.10.</b> El sistema nos lleva a la página principal en el menú usuarios.</p>	

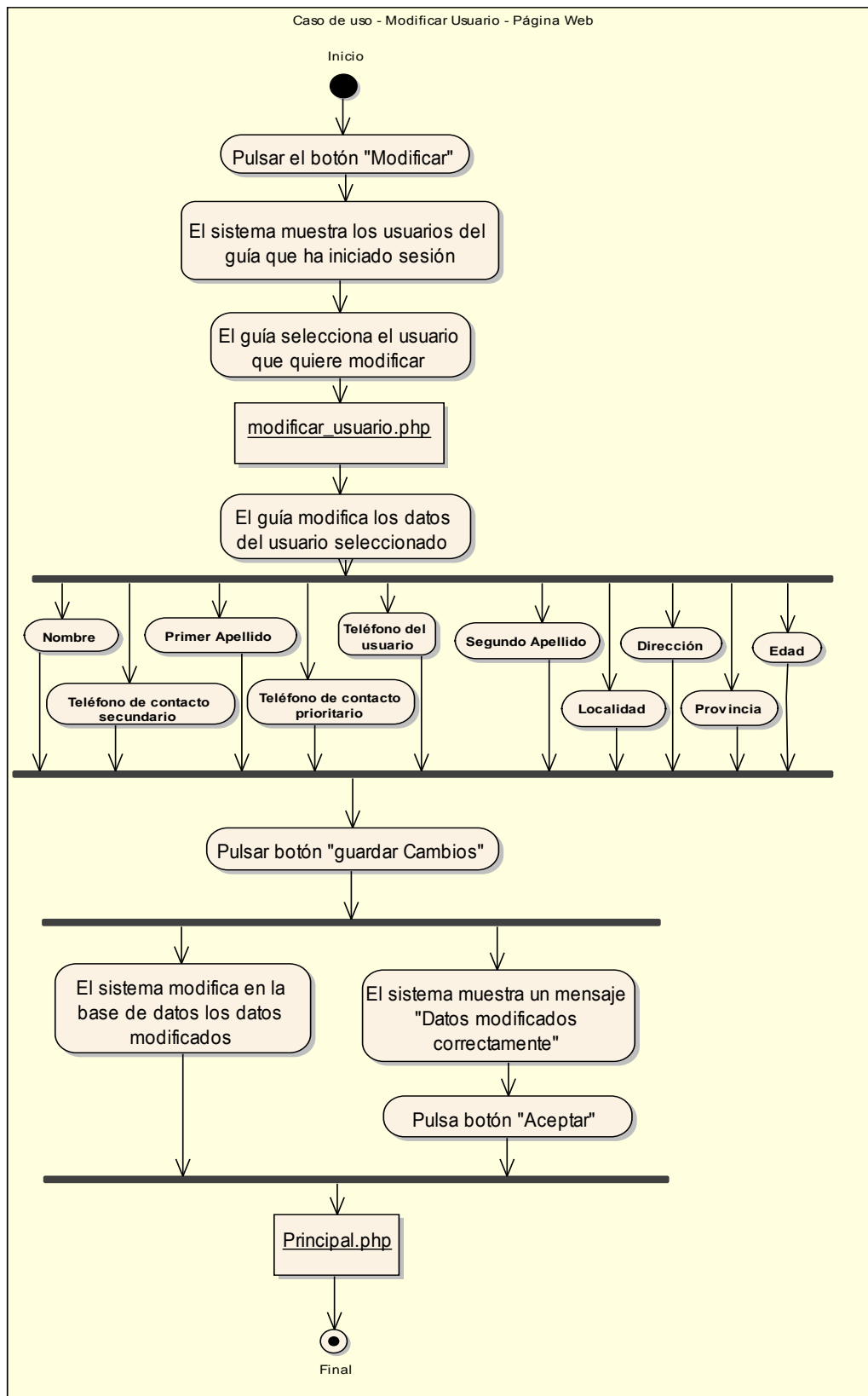


Ilustración 26: Diagrama de actividad del caso de uso Modificar Usuario de la página Web

## MODIFICAR RUTAS DE UN USUARIO

Tabla 14: Caso de uso: Modificar Rutas De Un Usuario - Aplicación WebTutor		
Descripción	Poder ver las rutas de un usuario seleccionado.	
Restricciones	Pre-Condiciones	Haber iniciado sesión en el sistema, estar en el menú rutas y haber pulsado el botón de usuario.
	Post-Condiciones	Visualizar las rutas del usuario.
Secuencia de eventos	<p><b>P.1.</b> Pulsar botón “usuario”.</p> <p><b>P.2.</b> El sistema nos muestra los usuarios del guía que ha iniciado sesión.</p> <p><b>P.3.</b> El guía selecciona el usuario, del cual quiere visualizar sus rutas.</p> <p><b>P.4.</b> El sistema busca en la bases de datos las rutas del usuario seleccionado.</p> <p><b>P.4.1.</b> El sistema encuentra rutas del usuario seleccionado.</p> <p><b>P.4.1.1.</b> El sistema muestra las rutas del usuario seleccionado.</p> <p><b>P.4.2.</b> EL sistema no encuentra rutas del usuario seleccionado.</p> <p><b>P.4.2.1.</b> El sistema muestra el mensaje “Aún no has introducido ninguna ruta”.</p>	

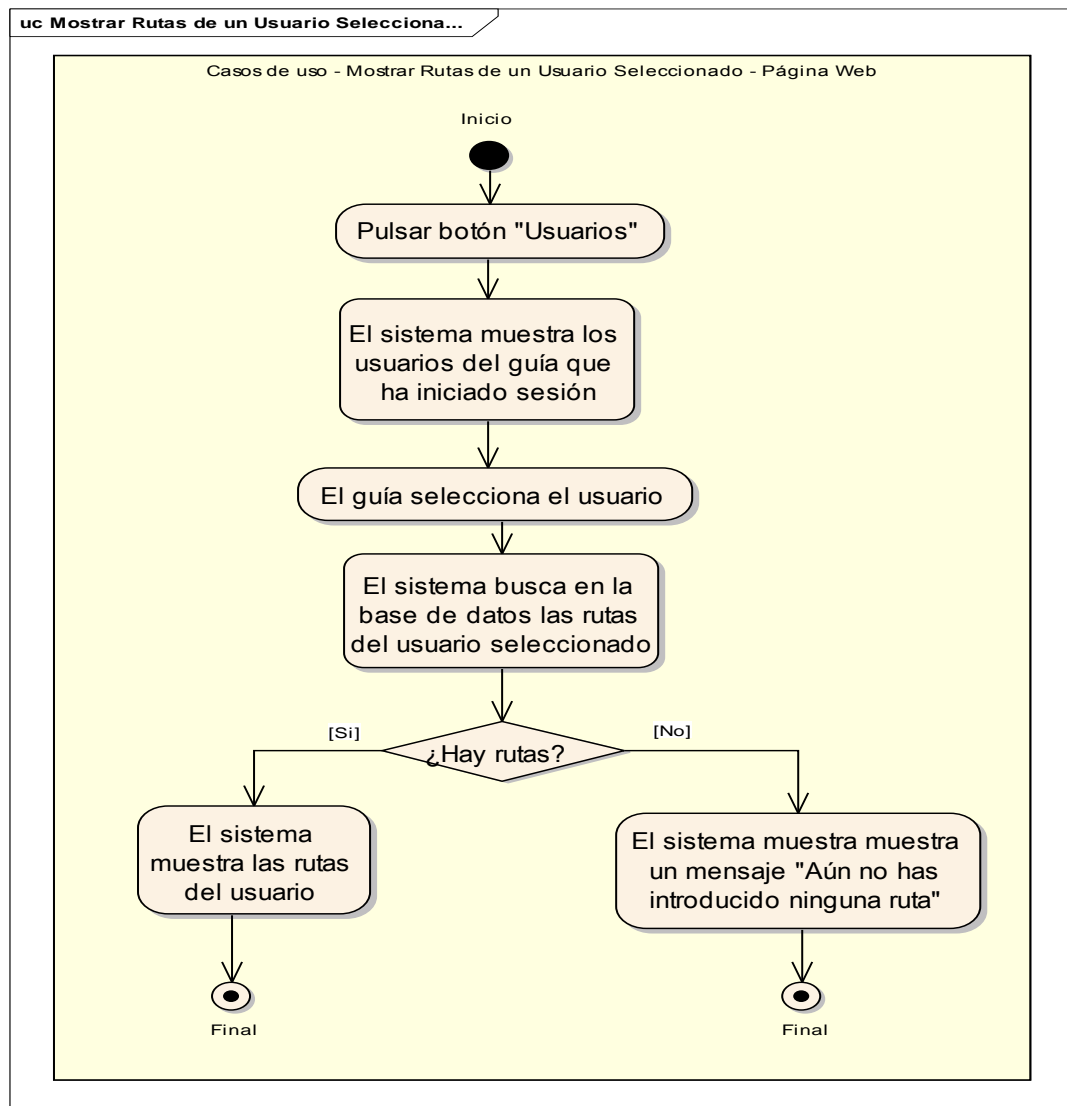


Ilustración 27: Diagrama de actividad del caso de uso Mostrar rutas de un usuario de la página Web



## REGISTRO

Tabla 15: Caso de uso: Registro - Aplicación WebTutor		
Descripción	Registro de un guía en el sistema.	
Restricciones	Pre-Condiciones	Tener creado una cuenta del correo y no estar registrado en el sistema
	Post-Condiciones	El guía queda registrado en el sistema
Secuencia de eventos	<p><b>P.1.</b> Introducir los datos requeridos para el registro (Nombre, Correo, Confirmación de correo, Contraseña y Confirmación de contraseña)</p> <p><b>P.2.</b> Pulsar botón “Registrar”</p> <p><b>P.2.1.</b> El sistema busca en la base de datos y verifica que el usuario no está registrado en el sistema.</p> <p><b>P.2.1.1.</b> Creación del nuevo guía en la base de datos.</p> <p><b>P.2.2.</b> El sistema busca en la base de datos y verifica que el usuario está registrado en el sistema.</p> <p><b>P.2.2.1.</b> El sistema muestra una ventana de Error, indicando que el usuario ya está registrado en el sistema.</p> <p><b>P.3.</b> El sistema nos lleva a página index.</p>	

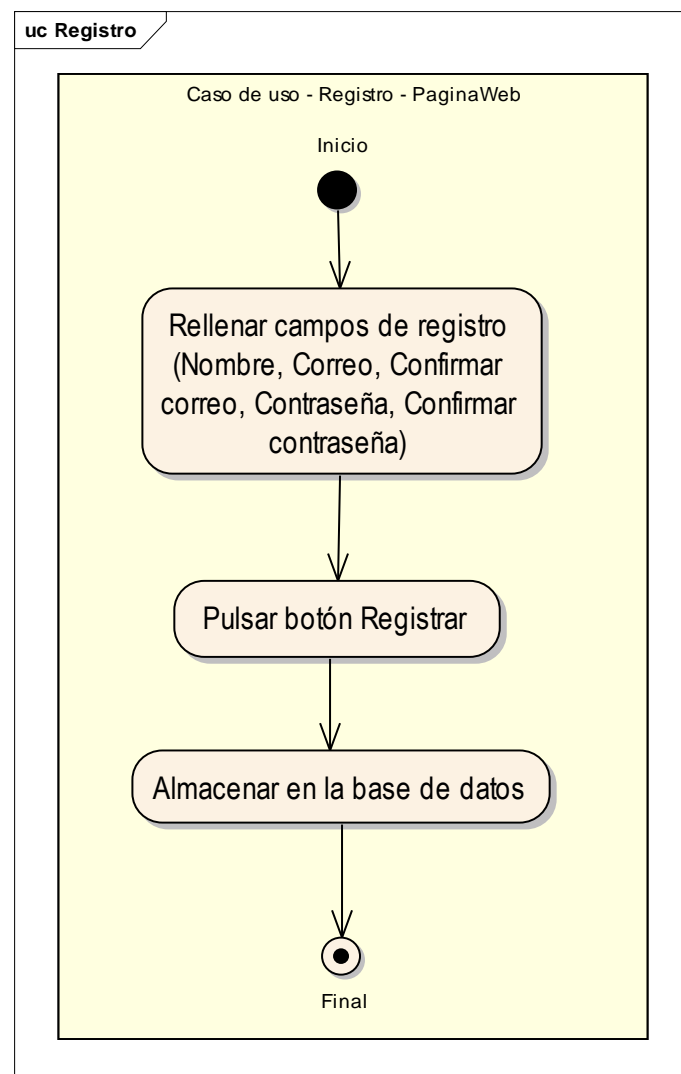


Ilustración 28: Diagrama de actividad del caso de uso Registro de la página Web

## VALORAR RUTA

Tabla 16: Caso de uso: Valorar Ruta - Aplicación WebTutor		
Descripción	El guía podrá marcar los puntos intermedios de la ruta y dar una valoración por aceras de los tramos.	
Restricciones	Pre-Condiciones	Haber iniciado sesión en el sistema, estar en el menú de rutas, tener creada la ruta que quiere valorar y pulsar en dicha ruta.
	Post-Condiciones	Valorar la ruta seleccionada.
Secuencia de eventos	<p><b>P.1.</b> Pulsar botón “Valorar” de la ruta.</p> <p><b>P.2.</b> El sistema nos lleva a la página valorar.php</p> <p><b>P.3.</b> El guía visualiza los puntos de inicio y final de ruta que previamente se introdujo cuando se creo la ruta y pulsa el botón “Configurar la ruta”.</p> <p><b>P.4.</b> El sistema muestra unas breves indicaciones para poder marcar los tramos que posteriormente se van a valorar.</p> <p><b>P.5.</b> El guía pulsa el botón “Aceptar”.</p> <p><b>P.6.</b> El sistema muestra una línea entre el punto de inicio y el punto de final de la ruta.</p> <p><b>P.7.</b> El guía deberá arrastrar la línea hacia los cruces o cambios de calles y marcar estos puntos, de inicio a final.</p> <p><b>P.8.</b> El guía pulsa el botón “Perfecto, vamos a valorar”.</p> <p><b>P.9.</b> El sistema nos muestra los botones con todos los tramos que anteriormente fueron seleccionados. Los botones se activarán en orden, del primero al último.</p> <p><b>P.10.</b> El guía pulsa el botón “Valorar tramo X”.</p> <p><b>P.10.1.</b> El sistema muestra el sistema de valoración de aceras (Izquierda y derecha) y el botón de valorar.</p> <p><b>P.10.2.</b> El guía valorará los factores mostrados mediante los iconos, para cada acera.</p> <p><b>P.10.3.</b> El guía pulsara el botón “Valorar”.</p> <p><b>P.11.</b> Una vez valorados todos los tramos el sistema mostrar el botón de guardar.</p> <p><b>P.12.</b> El guía pulsa el botón “Guardar”.</p> <p><b>P.13.</b> El sistema guarda en la base de datos todos los tramos y sus valoraciones asociándolos a la ruta.</p> <p><b>P.14.</b> El sistema nos lleva a la página principal en menú de rutas.</p>	

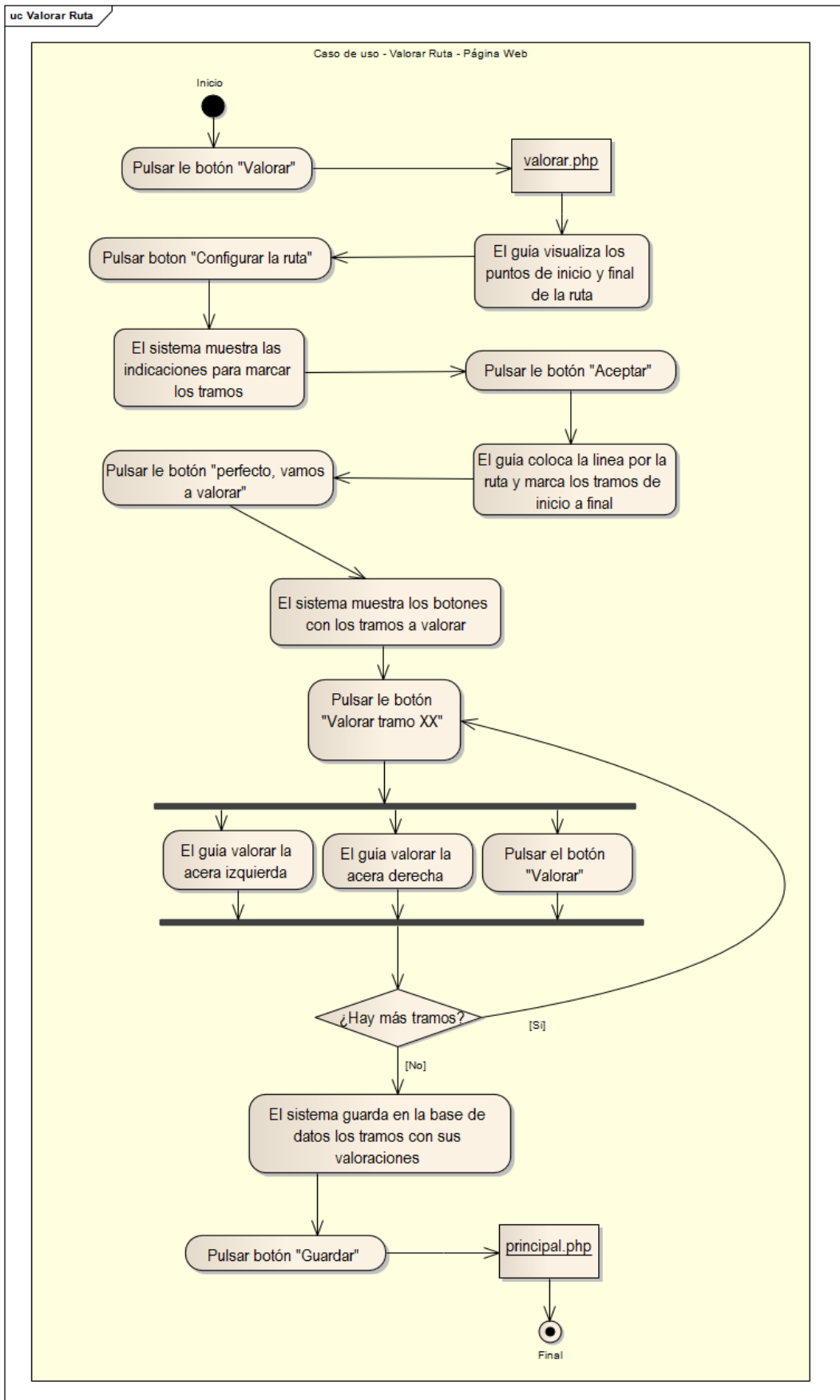
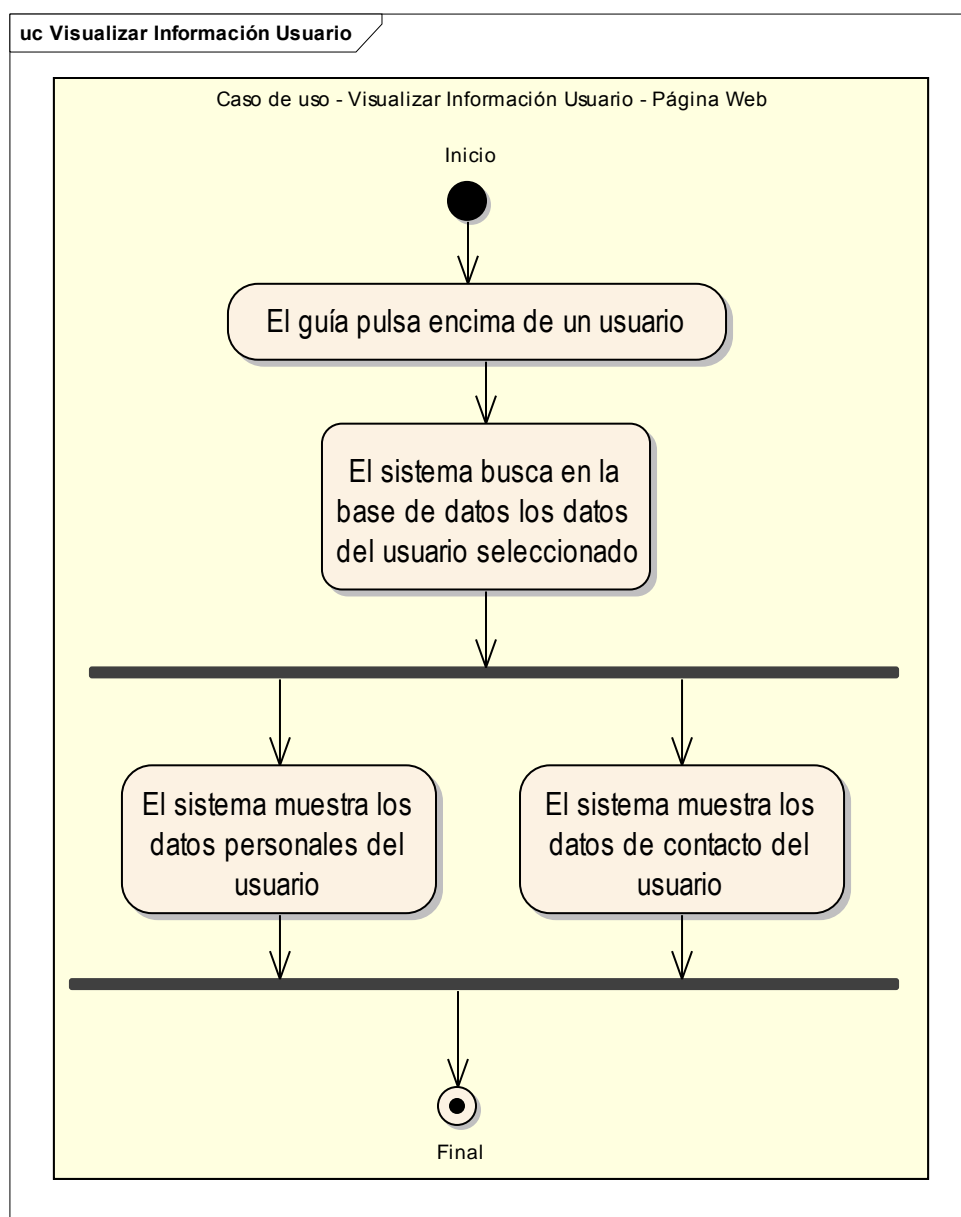


Ilustración 29: Diagrama de actividad del caso de uso Valorar Ruta de la página Web

## VISUALIZAR INFORMACIÓN DE UN USUARIO

**Tabla 17: Caso de uso: Visualizar Información de Un Usuario - Aplicación WebTutor**

Descripción	Poder visualizar los datos de un usuario.	
Restricciones	Pre-Condiciones	Haber iniciado sesión en el sistema, estar en el menú usuario y tener al menos un usuario.
	Post-Condiciones	Visualizar toda la información de un usuario.
Secuencia de eventos	<p><b>P.1.</b> El guía pulsa encima de un usuario.</p> <p><b>P.2.</b> El sistema busca en la base de datos todos los datos del usuario seleccionado.</p> <p><b>P.3.</b> El sistema muestra los datos personales del usuario.</p> <p><b>P.4.</b> El sistema muestra los datos de contactos del usuario.</p>	

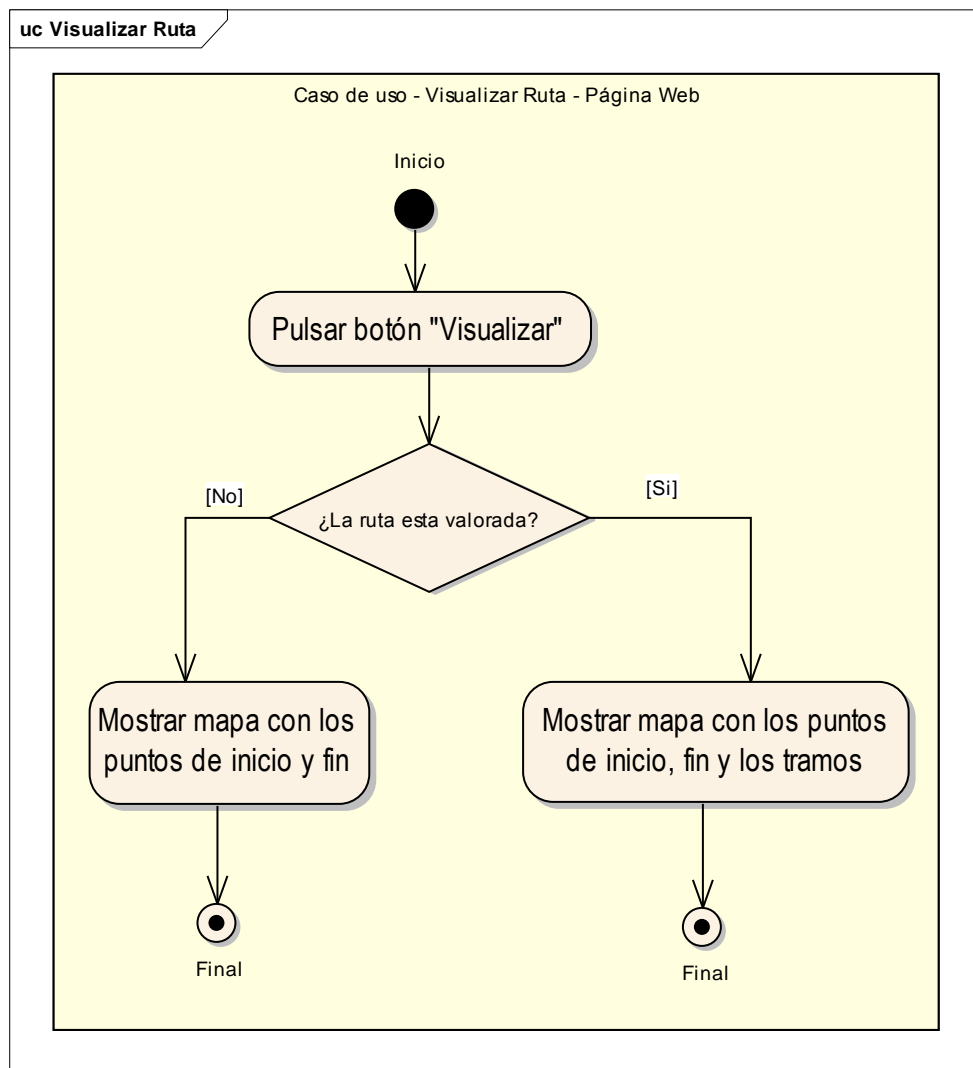


*Ilustración 30: Diagrama de actividad del caso de uso Visualizar información Usuario de la página Web*

## VISUALIZAR RUTAS

**Tabla 18: Caso de uso: visualizar Rutas - Aplicación WebTutor**

Descripción	Mostrar el mapa de la ruta seleccionada.	
Restricciones	Pre-Condiciones	Iniciar sesión en el sistema, haber pulsado rutas en el menú y pulsar botón visualizar de la ruta seleccionada.
	Post-Condiciones	El guía podrá visualizar el mapa de la ruta
Secuencia de eventos	<p><b>P.1.</b> Pulsar el botón “Visualizar”.</p> <p><b>P.2.</b> El sistema busca en la bases de datos las rutas del guía que inicio la sesión</p> <p><b>P.3.</b> El sistema indica si la ruta está valorada.</p> <p><b>P.3.1.</b> El sistema indica que la ruta está valorada.</p> <p><b>P.3.1.1.</b> El sistema muestra el mapa con los puntos de inicio, fin y los puntos que indican los tramos de la ruta.</p> <p><b>P.3.2.</b> El sistema indica que la ruta no está valorada</p> <p><b>P.3.2.1.</b> El sistema muestra el mapa con los puntos de inicio y fin de la ruta.</p>	



*Ilustración 31: Diagrama de actividad del caso de uso Visualizar Rutas de la página Web*

## Paquetes y clases

### Index.php

En esta página se encuentra el acceso de usuarios ya registrados y el registro de nuevos usuarios.

- *funcion\_valorar.js*: Es un archivo JavaScript en el que se valida el formulario de registro de un nuevo guía en el sistema, para ello hacemos uso de la función `validate` de `jquery`.

### Principal.php

En esta página podemos visualizar a todos los usuarios del guía (modificarlos, eliminarlos, visualizar la información de un usuario concreto o añadir usuarios nuevos) que previamente accedió al sistema, las rutas de sus usuarios (modificarlas, valorarlas, visualizar una ruta concreta, eliminarlas o añadir otra nueva) y la posibilidad de importar o exportar sus usuarios.

- *Escribir\_fichero.js*: Es un archivo JavaScript que se encarga de escribir en un archivo de texto la información de los usuarios que se quieran exportar.
  - Método `descargarArchivo(Object,String)`: Crea un archivo nuevo con el nombre dado y el link de descarga del archivo.
  - Método `generarTexto(String)`: Genera el texto que se va escribir en el archivo.
  - Método `obtenerDatos(Array)`: Obtiene la información de los usuarios que van a ser exportados.
- *Funcion\_validar.js*: Es un archivo JavaScript que se encarga de validar los campos del formulario de añadir nueva ruta y de mostrar y ocultar los campos de dicho formulario.
  - Método `validar(int)`: se encarga de validar que el campo del nombre, la calle de inicio, la calle de final de la ruta no estén vacíos y de ocultar el campo ya completado y mostrar el siguiente a completar.
  - Método `escribir(int)`: Se encarga de mostrar un mensaje de error en campo correspondiente.
  - Método `validar_menu_tipo_via(String,int)`: Se encarga de habilitar el campo en el que se ingresa las calles de inicio y de final cuando se halla seleccionado un tipo de vía.
  - Método `modificar_completo(int)`: Nos permite modificar el nombre de la calle de inicio y final de la ruta.
  - Método `modificar_marker(int)`: Habilita la posibilidad de modificar el marker de inicio y final de la ruta al punto deseado.
  - Método `modificado_completado(int)`: Una vez modificado el marker de inicio o final deshabilita la posibilidad de modificarlo.
  - Método `pasar_a_final()`: Oculta la parte del formulario de inicio de la ruta y muestra la parte del formulario de final de la ruta.

- Método `pasar_a_finalizar()`: Oculta la parte del formulario de final de la ruta y muestra los botones para finalizar y añadir o cancelar la nueva ruta.
  - Método `validar_menu_usuarios(String,int)`: Muestra el usuario seleccionado por el guía al que se le va asignar la nueva ruta.
- *leer\_fichero.js*: Es un archivo JavaScript en el que se lee el archivo recuperado de los usuarios que se van a importar.
  - Método `leer(int)`: Se encarga de sacar la información del archivo de los usuarios que se van a importar.
- *funciones\_botones.js*: Es un archivo JavaScript que se encarga de gestionar las acciones de los botones de la página principal.
  - Método `muestra_oculta_usuario(int,String,String,String,int,String,String,int,int,int)`: Se encarga de mostrar o ocultar la información de un usuario cuando pulsamos encima del mismo.
  - Método `mostrar_datos_izq(int)`: Muestra la información dependiendo de la parte del menú (usuarios, rutas, configuración) donde nos encontremos.
  - Método `mostrar_menu(int)`: Muestra la imagen correspondiente a la parte del menú (usuarios, rutas, configuración) donde nos encontremos.
  - Método `cambiar_a_rutas_usuario()`: Se encarga de mostrar las rutas del usuario seleccionado.
  - Método `abrir_alerta_eliminar(int,String,String)`: Se encarga de abrir el mensaje de eliminar con el nombre del usuario o el nombre de la ruta seleccionada.
  - Método `cerrar_alerta_eliminar(boolean,String)`: Se encarga de cerrar el mensaje de eliminar, eliminando el usuario o la ruta si se pulsó el botón “sí” o cerrando el mensaje de eliminar si se pulsó el botón “no”.
- *funciones\_mapa.js*: Es un archivo JavaScript que se encarga de gestionar todo lo relacionado con los mapas del menú de rutas de la página principal.
  - Método `initialize_mapa_modal_inicio()`: Inicializa el mapa de inicio de la ruta cuando estamos añadiendo una nueva ruta.
  - Método `marcar_inicio(String)`: Busca en Google Maps a través de geocoder las coordenadas de la calle introducida en el inicio de la ruta y de dibujarla en el mapa de inicio de ruta.
  - Método `initialize_mapa_modal_final()`: Inicializa el mapa de final de la ruta cuando estamos añadiendo una nueva ruta.
  - Método `marcar_final(String)`: Busca en Google Maps a través de geocoder las coordenadas de la calle introducida en el final de la ruta y de dibujarla en el mapa de final de ruta.
  - Método `activar_mover_marker(int)`: Se encarga de activar la propiedad del marker de un mapa para poder ser desplazado por el mapa de inicio final de la ruta.
  - Método `desactivar_mover_marker(int)`: Se encarga de desactivar la propiedad del marker de un mapa para poder ser desplazado por el mapa de inicio o final de la ruta.

- Método GuardarCoordenadasBBDD(): Se encarga de añadir a la base de datos la nueva ruta con el nombre de la ruta y las coordenadas de inicio y final de la ruta.
- Método guardar\_nombre\_ruta(String): Guarda en una variable el nombre de la ruta que estamos creando para después poder añadirla a la base de datos.
- Método guardar\_nombre\_inicio\_ruta(String): Guarda en una variable el nombre de la calle de inicio de la ruta que estamos creando para después poder añadirla a la base de datos.
- Método guardar\_nombre\_final\_ruta(String): Guarda en una variable el nombre de la calle de final de la ruta que estamos creando para después poder añadirla a la base de datos.
- Método guardar\_id\_usuario(int): Guarda en una variable el usuario de la ruta que estamos creando para después poder añadirla a la base de datos.
- Método muestra\_oculta(String,int,double,double,double,double): Se encarga de mostrar u ocultar el mapa de la ruta cuando pulsamos visualizar dicha ruta.
- Método initialize\_mapa\_ruta(double,double,double,double): Inicializa el mapa cuando pulsamos visualizar una ruta.
- Método marcar\_puntos(): Dibuja los marker de los tramos de la ruta si está valorada cuando pulsamos visualizar dicha ruta.
- Método marcar\_puntos\_de\_inicio\_final(): Dibuja los marker de inicio y final de la ruta sino está valorada cuando pulsamos visualizar dicha ruta.
- Método puntos\_de\_los\_tramos(String): Se encarga de sacar todas las coordenadas de los tramos de una ruta.
- Método eliminar\_punteros(): Se encarga de vaciar los array para que no se mezclen cuando pulsamos visualizar varias rutas.

### **Modificar\_ruta.php**

En esta página podemos modificar el nombre de una ruta y modificar las valoraciones de cada uno de los tramos de dicha ruta.

- *funciones\_botones.js*: Es un archivo JavaScript que se encarga de gestionar todos los botones de la página modificar ruta.
  - Método cambiar\_id(int): Guarda en una variable la ruta que estamos modificando para después poder modificar en la base de datos.
  - Método cambiar\_nombre(String): Guarda en una variable el nombre de la ruta que estamos modificando para después poder modificarla en la base de datos.
  - Método modificar\_nombreRuta(String): Se encarga de recoger el nuevo nombre de la ruta que estamos modificando y modificarla en la base de datos.
  - Método dame\_puntos\_valorar(String): Muestra en el mapa los marker de inicio, final y tramos de la ruta.
  - Método mostrar\_valorar\_tramos(int,boolean): Muestra en el mapa el tramo seleccionado y los factores correspondiente a ese tramo.
  - Método factores(int): Muestra los factores de las dos aceras de un tramo.



- Método `factores_izquierda(int)`: Muestra los factores de la acera izquierda de un tramo.
  - Método `factores_derecha(int)`: Muestra los factores de la acera derecha de un tramo.
  - Método `guardar_valor_factor(int,int,int,String)`: Se encarga de colorear la valoración pulsada.
  - Método `desmarcar_todas_las_caras(int,int,String)`: se encarga de descolorar todas las valoraciones.
  - Método `inicializar_matriz_puntuaciones()`: Se encarga de inicializar la matriz donde guardaremos las puntuaciones marcadas por el guía.
  - Método `actualizar_todos_los_tramos(String)`: Se encargar de guardar en la matriz de puntuaciones todas las valoraciones modificadas por el guía y posteriormente actualizar la base de datos.
- *funciones\_mapa.js*: Es un archivo JavaScript que se encarga de gestionar todo lo relacionados con el mapa para modificar la ruta.
    - Método `initialize(String)`: Inicializa el mapa para modificar la ruta, dibujando la ruta completa con los marker de inicio, final y todos sus tramos.
    - Método `sacar_datos(String)`: Se encarga de sacar los datos de la ruta.
    - Método `sacar_coordenadas(String)`: Se encarga de sacar todas las coordenadas de los tramos de la ruta.
    - Método `sacar_marker(array)`: Dadas unas coordenadas se encarga de dibujar en el mapa los markers de esas coordenadas.
    - Método `dibujar_tramos()`: Se encarga de dibujar la línea que une los markers de una ruta.

## Valorar.php

En esta página podemos añadir los tramos de una ruta y valorar cada tramo por aceras.

- *Funciones\_mapa.js*: es un archivo JavaScript que se encarga de gestionar todo lo relacionado con el mapa de la página valorar.
  - Método `initialize(String)`: Inicializa el mapa cuando estamos valorando la ruta añadiendo el marker de inicio y de final de la ruta.
  - Método `anadir_puntos_valorar()`: Se encarga hacer una polilínea entre los marker de inicio y final de la ruta, hace que la polilínea sea editable y añade un evento que permite añadir nuevos marker para marcar los tramos.
  - Método `anadir_puntos_ruta(google.maps.Event)`: Permite añadir y dibujar los nuevos markers insertados por el guía en la valoración de la ruta.
  - Método `dejar_de_construir_la_ruta()`: Se encarga de hacer que la polilínea no sea editable.
  - Método `dame_puntos()`: Se encarga de devolver todas las coordenadas de los tramos de la ruta.

- *Funciones\_botones.js*: Es un archivo JavaScript que se encarga de gestionar todas las acciones de los botones de la página valorar
  - Método *pasar\_a\_menu2()*: Se encarga de pasar del menú donde se explica cómo construir la ruta al menú en que el guía debe marcar los markers en los cruces o cambios de calles.
  - Método *pasar\_a\_menu3()*: Se encarga de mostrar el menú donde se pueden valorar cada uno de los tramos por aceras.
  - Método *dame\_puntos\_valorar()*: Se encarga de mostrar los botones de los tramos que tiene la ruta anteriormente construida para posteriormente valorar.
  - Método *inicializar\_matriz\_puntuaciones()*: Se encarga de inicializar la matriz donde vamos a guardar las puntuaciones de cada tramo y cada acera.
  - Método *factores(int)*: Se encarga de mostrar el menú donde encontramos los factores que vamos a valorar para cada tramo
  - Método *factores\_izquierda(int)*: Se encarga de mostrar los factores de la acera izquierda del tramo.
  - Método *factores\_derecha(int)*: Se encarga de mostrar los factores de la acera derecha del tramo.
  - Método *guardar\_valor\_factor(int, int, int, String)*: Se encarga de guardar la valoración que el guía ha marcado en la ruta para cada factor y cada acera en la matriz de puntuaciones.
  - Método *desmarcar\_todas\_las\_caras(int, int, String)*: Se encargar de desmarcar todos los iconos de la valoración para seguir valorando otro factor.
  - Método *mostrar\_todas\_las\_caras(int, int, String)*: Se encarga de mostrar los iconos para poder valorar cada tramo.
  - Método *guardar\_tramos(int, int)*: Se encarga de guardar las coordenadas de los tramos.
  - Método *guardar\_todos\_los\_tramos()*: Se encarga de calcular la valoración total de toda la ruta con las valoraciones de cada tramo.

## **Usuario.php**

En esta página podemos añadir usuarios al sistema.

- *Funcion\_validar\_usuario.js*: Es un archivo JavaScript en el que se valida el formulario de añadir usuario en el sistema, para ello hacemos uso de la función *validate* de jquery.

## **Modificar\_usuario.php**

En esta página podemos modificar toda la información (Nombre, Apellidos, Dirección (Calle, Localidad, Provincia), Teléfonos y Edad) del usuario.

## 4.5 Arquitectura de la Aplicación del usuario (MiTutor)

El propósito de esa sección es explicar las diferentes funcionalidades de la aplicación del usuario a través de casos de uso y diagramas de actividades. Además se analizan las diferentes clases utilizadas en esta aplicación, dando una breve explicación para cada una de ellas.

### Casos de Uso

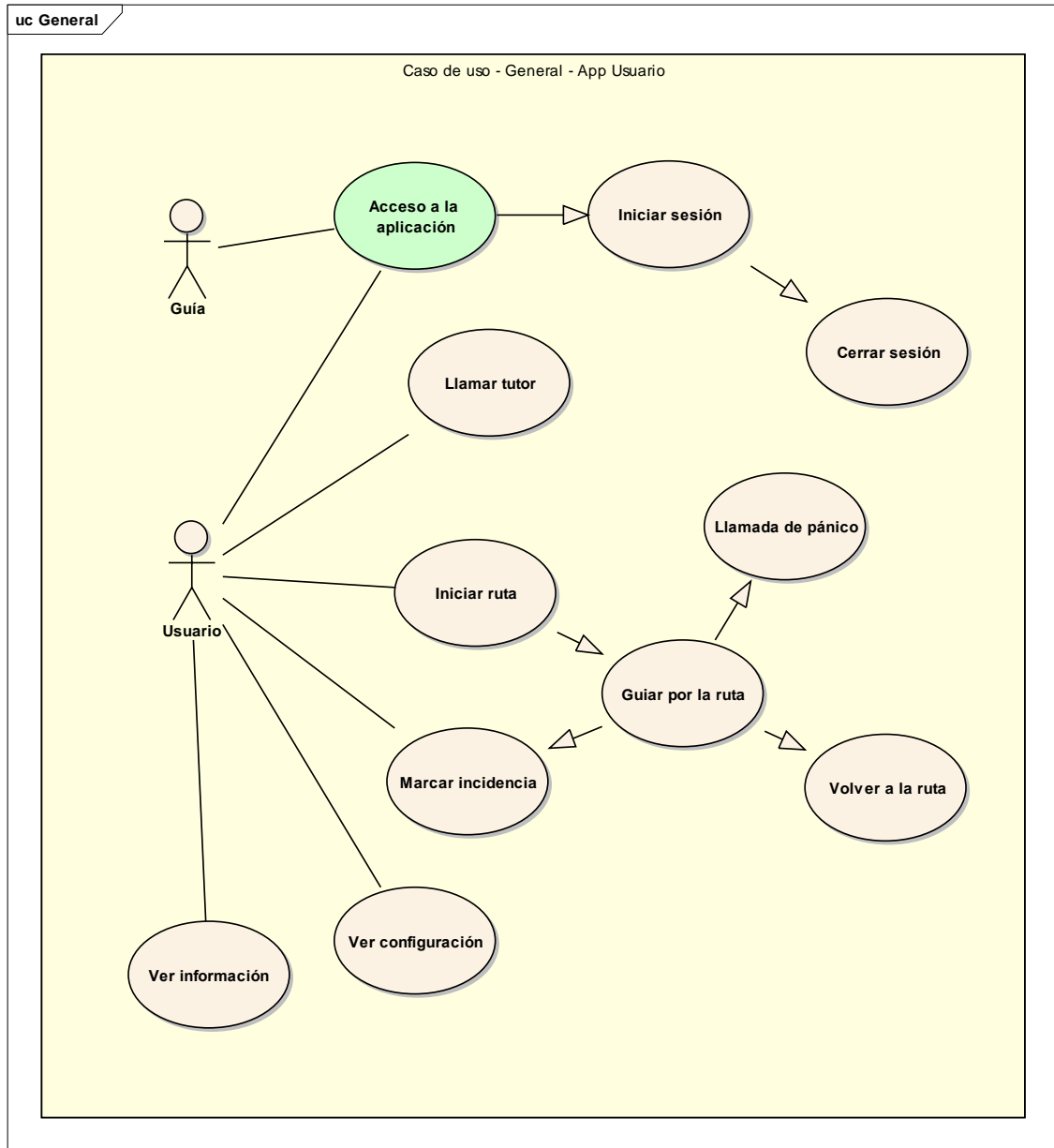


Ilustración 32: Diagrama general de casos de uso de la aplicación del usuario

## CERRAR SESIÓN

Tabla 19: Caso de uso: Cerrar Sesión - Aplicación MiTutor		
Descripción	El usuario querrá cerrar la sesión de la aplicación.	
Restricciones	Pre-Condiciones	El usuario debe haber iniciado la aplicación y encontrarse en el menú principal.
	Post-Condiciones	El usuario habrá cerrado la sesión de la aplicación.
Secuencia de eventos	<p><b>P.1.</b> El usuario pulsara el botón "Salir" o bien el botón "atrás" del terminal.</p> <p><b>P.2.</b> El sistema mandara un mensaje al usuario para confirmar la acción.</p> <p><b>P.2.1.</b> El usuario pulsa el botón "Si".</p> <p><b>P.2.1.1.</b> El sistema cerrara la aplicación.</p> <p><b>P.2.2.</b> El usuario pulsa el botón "No".</p> <p><b>P.2.2.1.</b> El sistema regresara al menú principal.</p>	

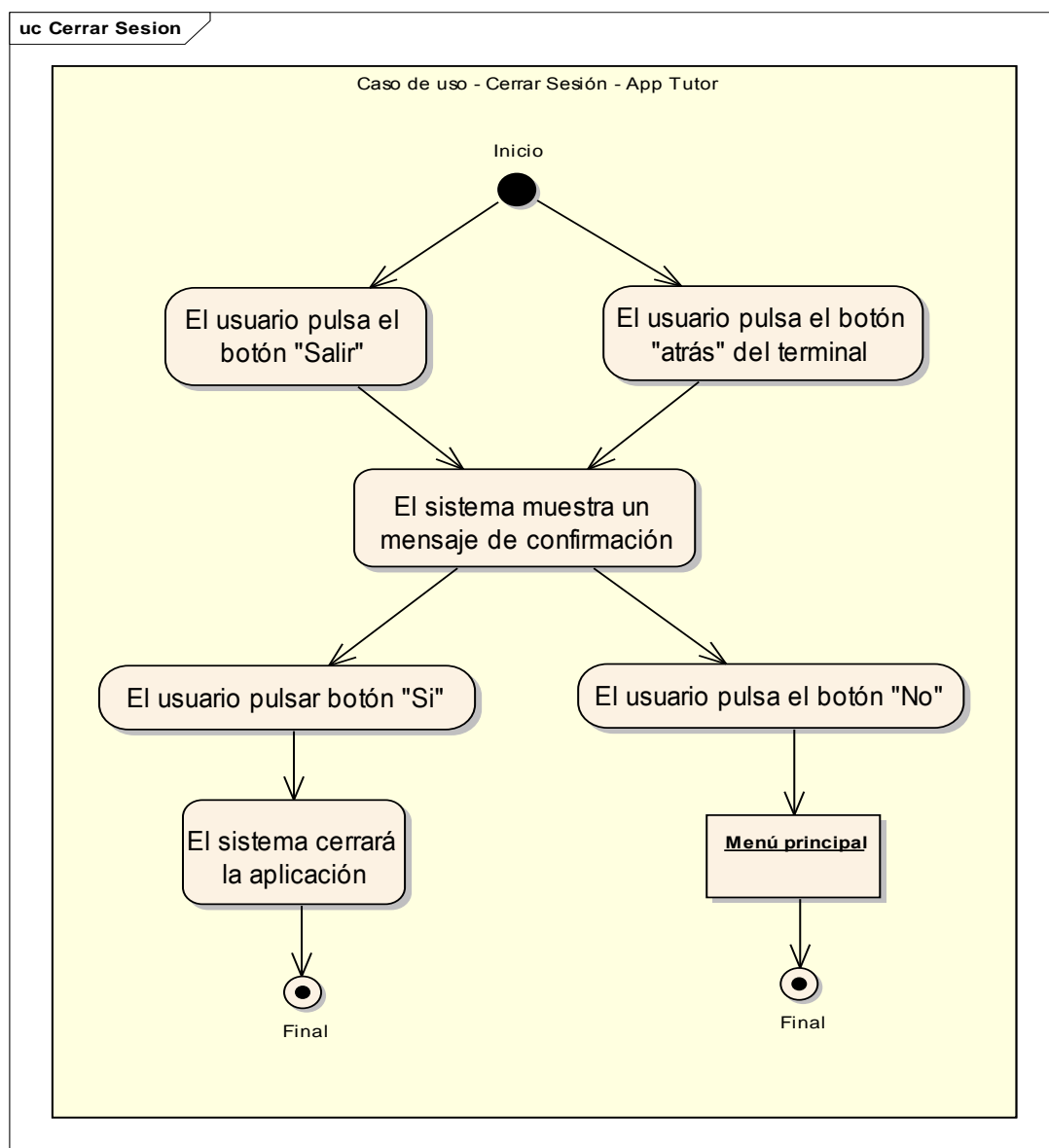


Ilustración 33: Diagrama de actividad del caso de uso cerrar sesión de la aplicación del usuario

## GUIAR POR LA RUTA

Tabla 20: Caso de uso: Guiar Por La Ruta - Aplicación MiTutor		
Descripción	El usuario será guiado durante la ruta indicándole en todo momento por qué acera debe ir para llegar por el camino optimo a su destino.	
Restricciones	Pre-Condiciones	El usuario debe haber iniciado la aplicación y encontrarse en el menú principal o bien en una ruta ya iniciada.
	Post-Condiciones	El usuario habrá llegado a su destino por las aceras mejor puntuadas por tu guía.
Secuencia de eventos	<p><b>P.1.</b> El sistema almacena el punto de inicio por un posible retorno a el.</p> <p><b>P.1.1.</b> El usuario tiene el GPS activado.</p> <p><b>P.1.1.1.</b> El sistema consulta cuál es la mejor acera para el tramo en el que está.</p> <p><b>P.1.1.2.</b> El sistema muestra al usuario por qué acera debe ir.</p> <p><b>P.1.1.3.</b> El sistema va actualizando la posición del usuario.</p> <p><b>P.1.1.4.</b> El sistema localiza el cambio de tramo.</p> <p><b>P.1.1.5.</b> El sistema almacena la localización para un posible retorno y volvemos a P.1.1.1.</p> <p><b>P.1.2.</b> El usuario no tiene activado el GPS</p> <p><b>P.1.2.1.</b> El sistema le mostrara un mensaje para activar la localización GPS.</p> <p><b>P.1.2.1.1.</b> El usuario pulsa "Si".</p> <p><b>P.1.2.1.1.1.</b> El sistema llevara al usuario a los ajustes para que active el GPS.</p> <p><b>P.1.2.1.2.</b> El usuario pulsa "No".</p>	

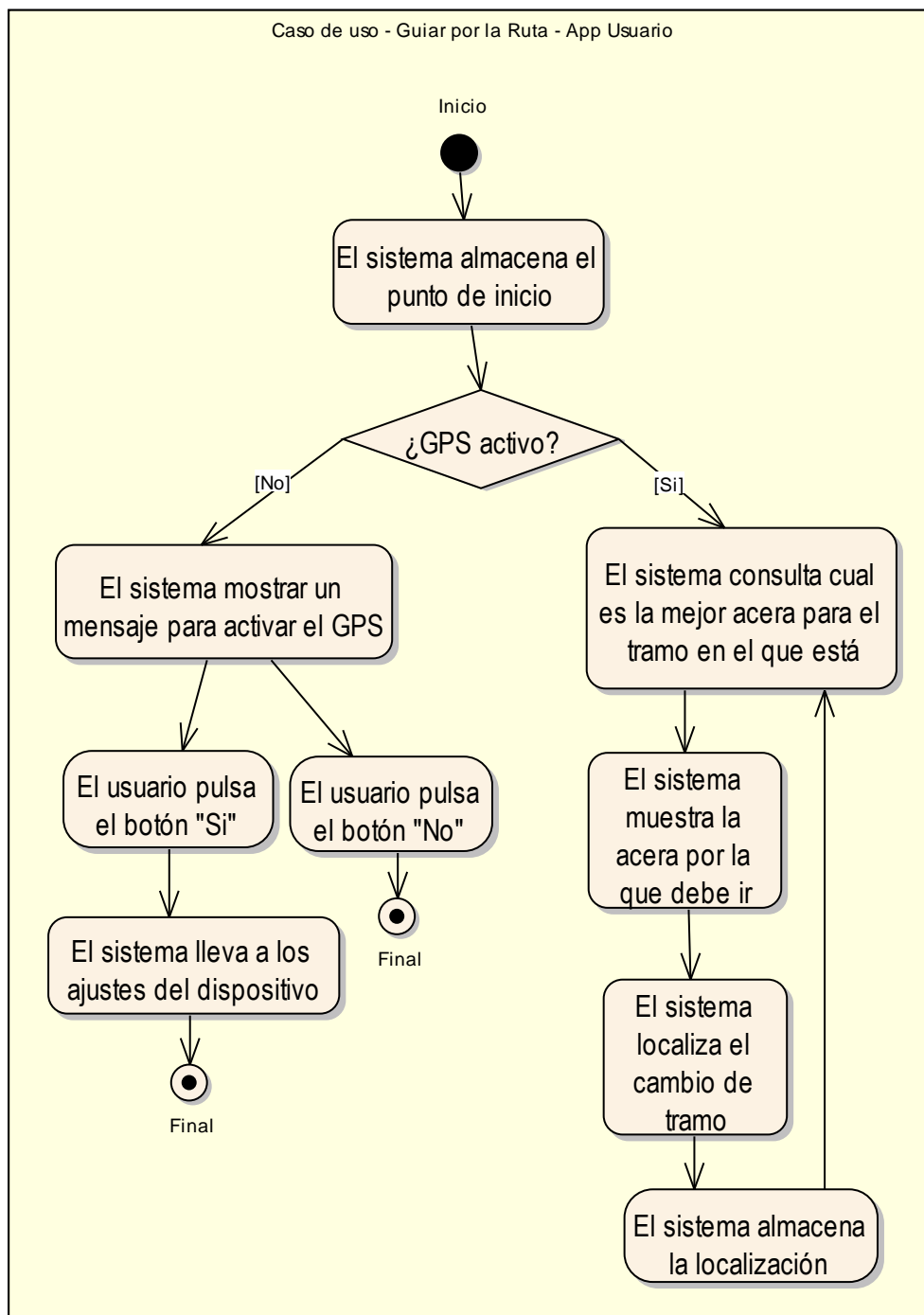


Ilustración 34: Diagrama de actividad del caso de uso guiar por la ruta de la aplicación del usuario

## INICIAR RUTA

Tabla 21: Caso de uso: Iniciar ruta - Aplicación MiTutor		
Descripción	El usuario podrá iniciar sus rutas.	
Restricciones	Pre-Condiciones	El usuario debe haber iniciado la aplicación y encontrarse en el menú principal.
	Post-Condiciones	El usuario habrá iniciado la ruta deseada y podrá ser guiado.
Secuencia de eventos	<p><b>P.1.</b> El usuario pulsara el botón "Iniciar Ruta".</p> <p><b>P.2.</b> El sistema buscara las rutas asociadas al usuario en la base de datos.</p> <p><b>P.2.1.</b> El sistema encuentra rutas.</p> <p><b>P.2.1.1.</b> El sistema mostrara un listado con todas las rutas.</p> <p><b>P.2.1.2.</b> El usuario seleccionara la ruta que desea iniciar.</p> <p><b>P.2.1.3.</b> El sistema consulta en la base de datos los tramos relacionados con la ruta seleccionada.</p> <p><b>P.2.1.3.1.</b> La ruta está valorada.</p> <p><b>P.2.1.3.1.1.</b> El sistema mostrara la ruta visualizando la información de los tramos, inicio y fin de la ruta.</p> <p><b>P.2.1.3.2.</b> La ruta no está valorada.</p> <p><b>P.2.1.3.2.1.</b> El sistema mostrara la ruta visualizando la información de inicio y final de la ruta.</p> <p><b>P.2.1.3.2.2.</b> El sistema muestra el mensaje “La ruta seleccionada no tiene tramos definidos”.</p> <p><b>P.2.2.</b> El sistema no encuentra rutas.</p> <p><b>P.2.2.1.</b> El sistema muestra el mensaje "Ninguno de sus usuarios tiene rutas definidas".</p>	

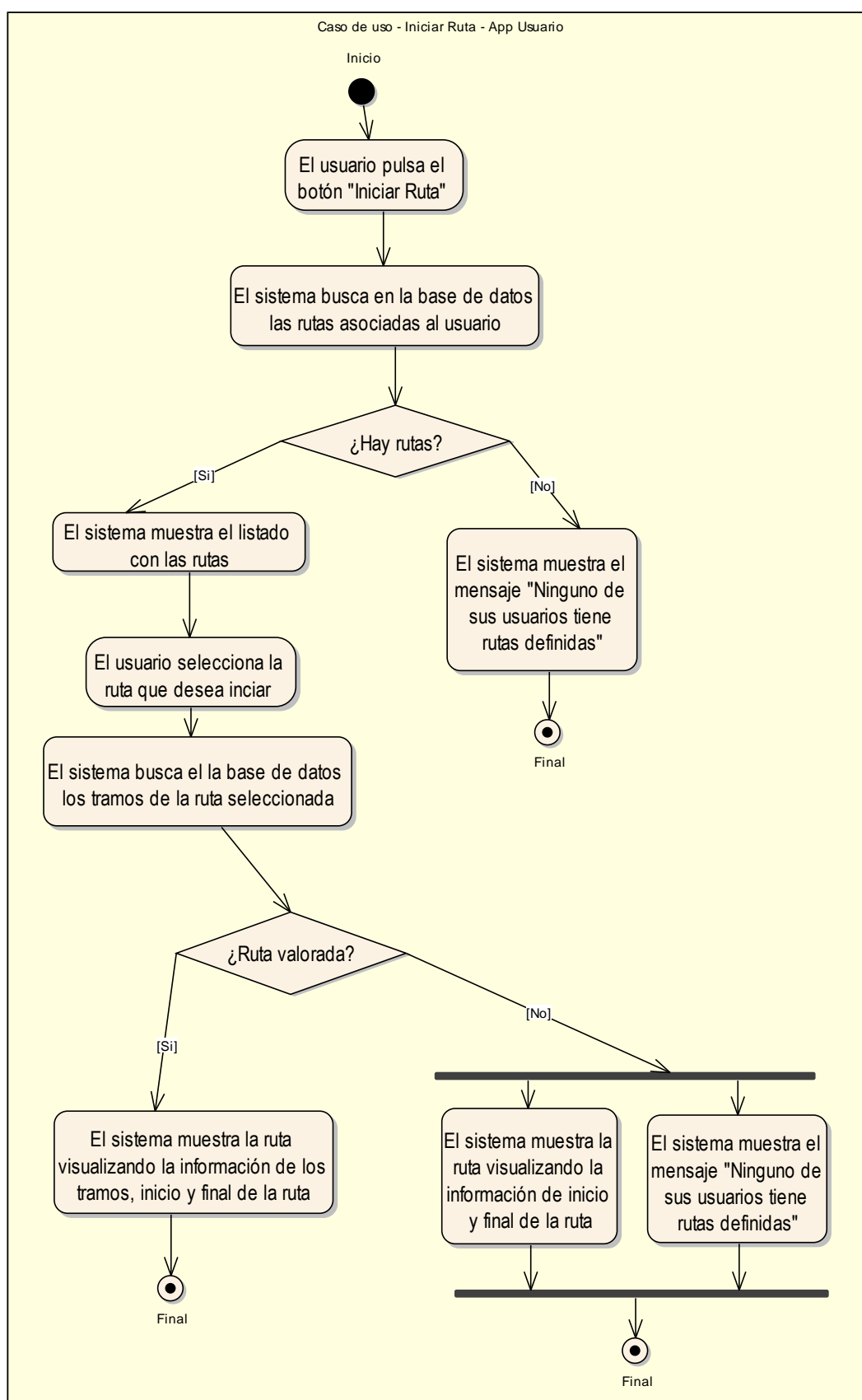


Ilustración 35: Diagrama de actividad del caso de uso iniciar ruta de la aplicación del usuario



## INICIAR SESIÓN

Tabla 22: Caso de uso: Iniciar Sesión - Aplicación MiTutor		
Descripción	Acceso a la aplicación MiTutor por del usuario por medio del guía.	
Restricciones	Pre-Condiciones	El guía debe de estar registrado en la base de datos y disponer de usuarios a su cargo.
	Post-Condiciones	Quedara iniciada la sesión del usuario para el posterior uso de las funcionalidades de la aplicación.
Secuencia de eventos	<p><b>P.1.</b> El sistema busca si los datos de acceso están ya guardados en la memoria del teléfono.</p> <p><b>P.1.1.</b> El sistema encuentra al guía en los datos de la aplicación.</p> <p><b>P.1.1.1.</b> El sistema nos lleva al menú principal de la aplicación.</p> <p><b>P.1.2.</b> El sistema no encuentra al guía en los datos de la aplicación.</p> <p><b>P.1.2.1.</b> El sistema nos lleva a la pantalla de inicio de sesión de la aplicación.</p> <p><b>P.1.2.2.</b> El usuario introducirá los datos requeridos por la aplicación para el inicio de sesión (Correo y Contraseña).</p> <p><b>P.1.2.3.</b> El usuario pulsara el botón "Entrar".</p> <p><b>P.1.2.3.1.</b> El sistema busca los datos de acceso en a base de datos.</p> <p><b>P.1.2.3.1.1.</b> El sistema encuentra al guía en base de datos.</p> <p><b>P.1.2.3.1.1.1.</b> El sistema consulta en la base de datos los usuarios del guía.</p> <p><b>P.1.2.3.1.1.2.</b> El sistema nos muestra los usuarios a cargo de ese guía.</p> <p><b>P.1.2.3.1.1.3.</b> El usuario selecciona el usuario que utilizara la aplicación.</p> <p><b>P.1.2.3.1.1.4.</b> El sistema guardara la información del guía en la memoria del teléfono para futuros accesos.</p> <p><b>P.1.2.3.1.1.5.</b> El sistema nos llevara al menú principal de la aplicación.</p> <p><b>P.1.2.3.1.2.</b> El sistema no encuentra al guía en la base de datos.</p> <p><b>P.1.2.3.1.2.1.</b> La aplicación nos informara "Los datos de acceso no son correctos".</p>	

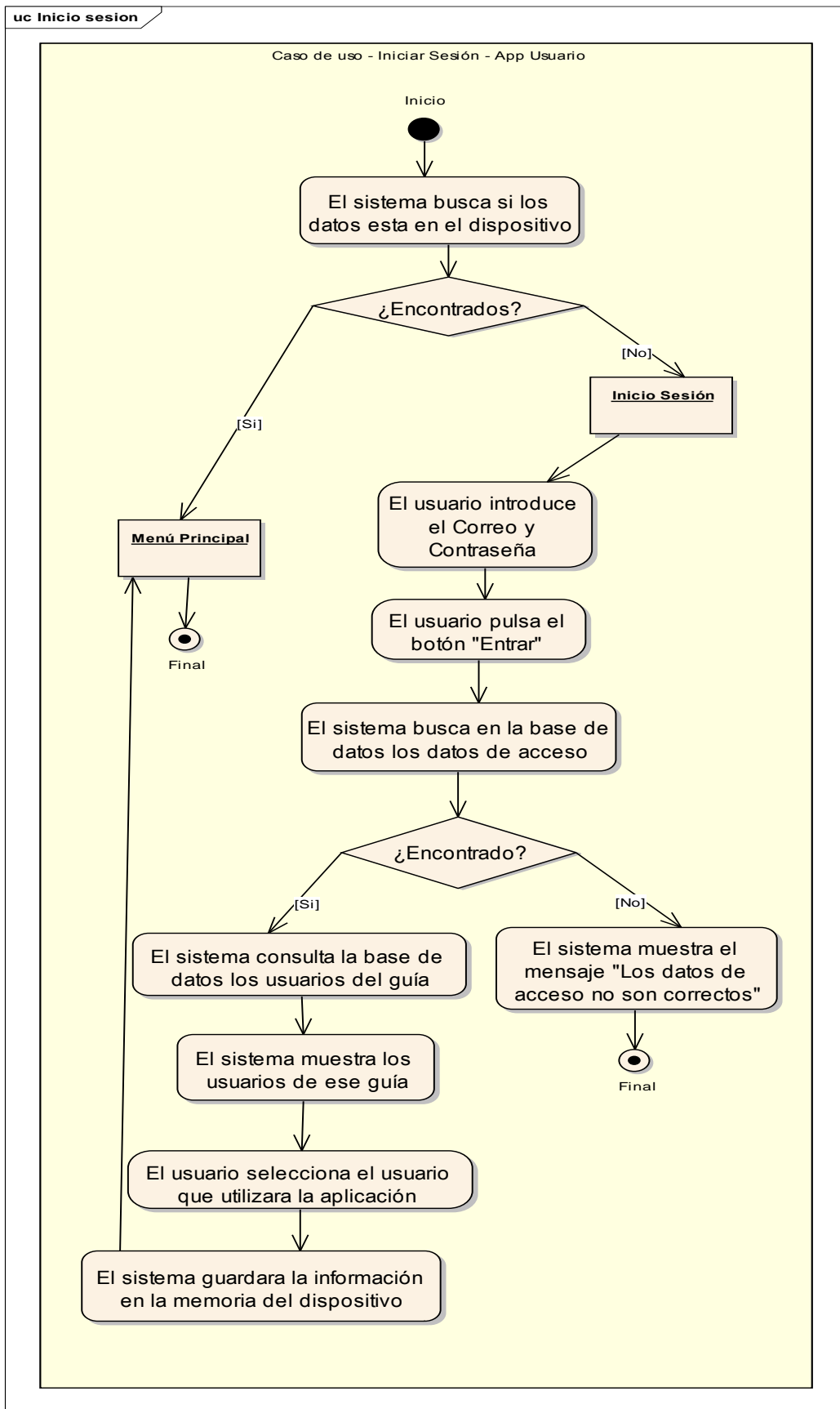
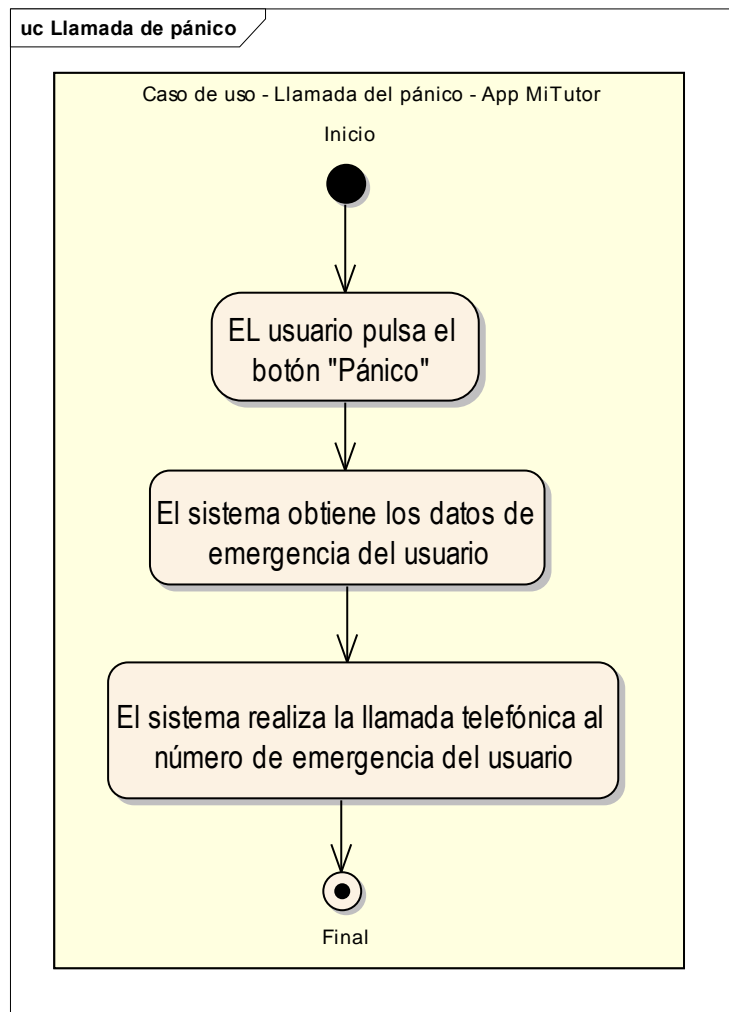


Ilustración 36: Diagrama de actividad del caso de uso iniciar sesión de la aplicación del usuario

# LLAMADA DE PÁNICO

Tabla 23: Caso de uso: Llamada De Pánico - Aplicación MiTutor		
Descripción	El usuario realiza una llamada telefónica a su contacto de emergencia.	
Restricciones	Pre-Condiciones	El usuario deberá haber iniciado una ruta y encontrarse en una situación de emergencia.
	Post-Condiciones	El usuario proseguirá con su ruta tras resolver su emergencia.
Secuencia de eventos	<p><b>P.1.</b> El usuario pulsa el botón “Pánico”.</p> <p><b>P.2.</b> El sistema obtiene los datos del usuario, donde se encuentran los números de emergencia.</p> <p><b>P.3.</b> El sistema realiza la llamada telefónica al número de teléfono de emergencia que tenía configurado el usuario.</p>	



*Ilustración 37: Diagrama de actividad del caso de llamada de pánico de la aplicación del usuario*

LLAMAR TUTOR

Tabla 24: Caso de uso: Llamar Tutor - Aplicación MiTutor		
Descripción	El usuario realiza una llamada telefónica a su tutor.	
Restricciones	Pre-Condiciones	El usuario deberá haber iniciado la aplicación y encontrarse en el menú principal.
	Post-Condiciones	
Secuencia de eventos	<b>P.1.</b> El usuario pulsa el botón “Llamada tutor”. <b>P.2.</b> El sistema obtiene los datos del tutor. <b>P.3.</b> El sistema realiza la llamada telefónica al número de teléfono del tutor.	

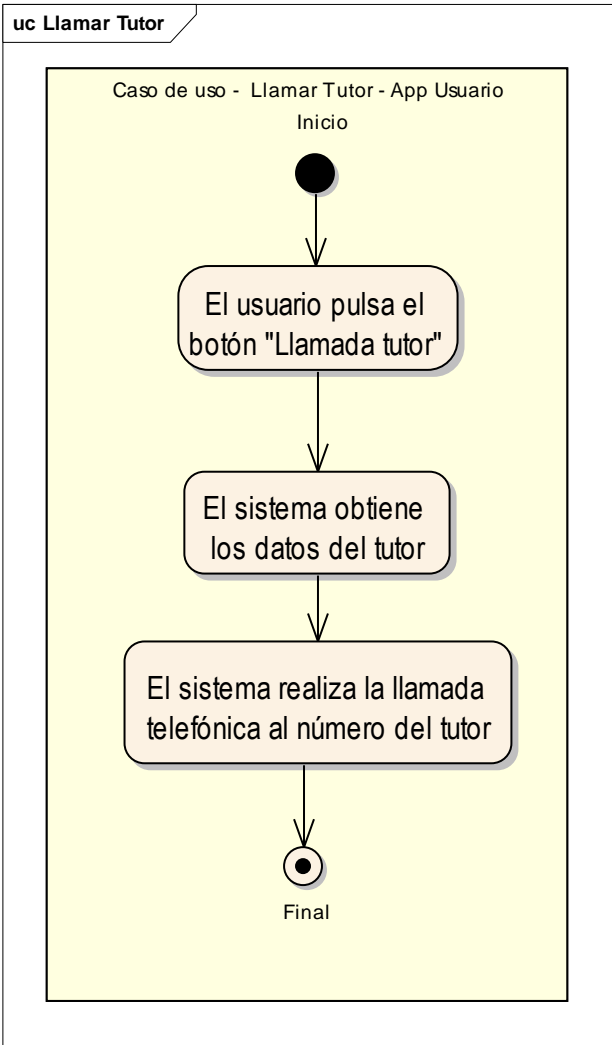


Ilustración 38: Diagrama de actividad del caso de uso Llamar tutor de la aplicación del usuario

VER CONFIGURACIÓN

Tabla 25: Caso de uso: Ver Configuración - Aplicación MiTutor		
Descripción	El usuario consultara y podrá modificar su configuración de la aplicación.	
Restricciones	Pre-Condiciones	El usuario deber haber iniciado la aplicación y encontrarse en el menú principal.
	Post-Condiciones	El usuario habrá comprobado su configuración de la aplicación.
Secuencia de eventos	<b>P.1.</b> El usuario pulsara el botón "Configuración". <b>P.2.</b> El sistema mostrara la interfaz de configuración de la aplicación.	

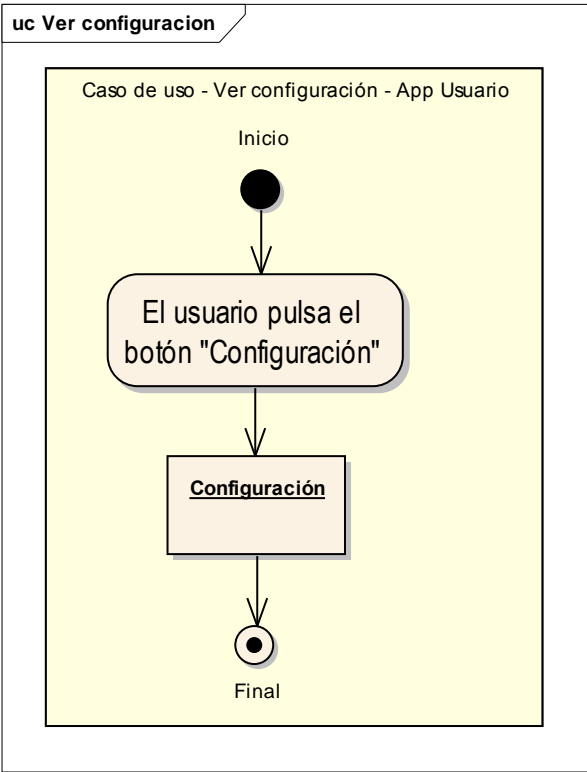


Ilustración 39: Diagrama de actividad del caso de uso Ver configuración de la aplicación del usuario

## MARCAR INCIDENCIA

Tabla 26: Caso de uso: Marcar Incidencia - Aplicación MiTutor		
Descripción	El usuario reportara una incidencia para que su tutor pueda encargarse de resolverla.	
Restricciones	Pre-Condiciones	El usuario debe haber iniciado la aplicación y encontrarse en el menú principal o bien en una ruta ya iniciada.
	Post-Condiciones	El usuario habrá reportado la incidencia.
Secuencia de eventos	<p><b>P.1.</b> El usuario pulsara el botón "Incidencias".</p> <p><b>P.2.</b> El sistema le llevara a una interfaz con un mapa para que el usuario marque el tipo de incidencia que quiere reportar.</p> <p>    <b>P.2.1.</b> El usuario no tiene activado el GPS.</p> <p>    <b>P.2.2.</b> El sistema le mostrara un mensaje para activar la localización GPS.</p> <p>        <b>P.2.2.1.</b> El usuario pulsa "Si".</p> <p>            <b>P.2.2.1.1.</b> El sistema llevara al usuario a los ajustes para que active el GPS.</p> <p>        <b>P.2.2.2.</b> El usuario pulsa "No".</p> <p><b>P.3.</b> El usuario selecciona la incidencia a reportar, utilizando los botones de cada tipo de incidencia (obstáculo, inseguridad, otros).</p> <p>    <b>P.3.1.</b> El usuario no tiene activado el GPS, retorno a P.2.2.</p> <p><b>P.4.</b> El sistema manda una notificación con la incidencia al guía del usuario.</p> <p><b>P.5.</b> El sistema guarda en la base de datos la incidencia reportada por el usuario.</p> <p><b>P.6.</b> El sistema nos llevara al menú principal o a la ruta.</p>	

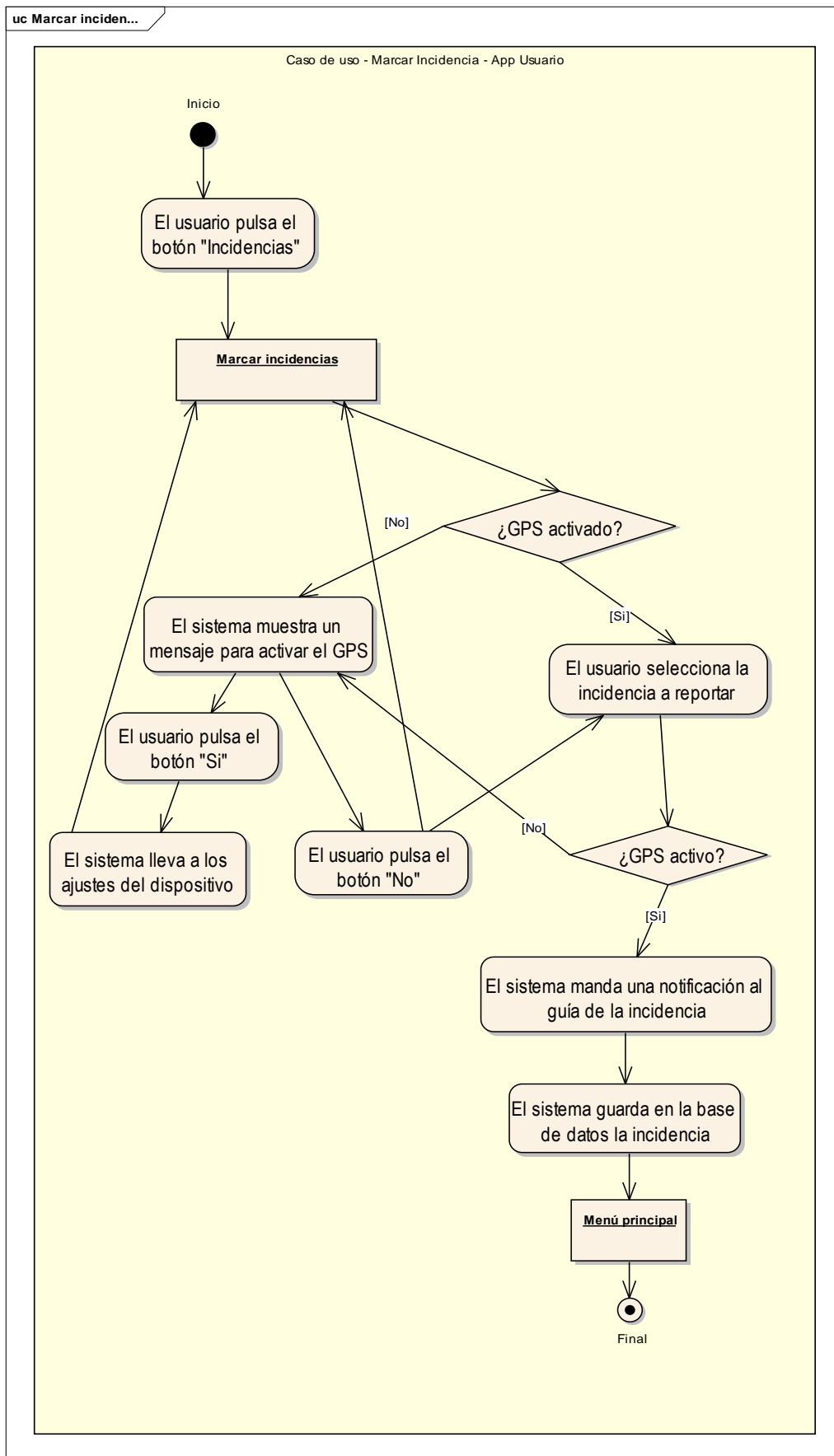


Ilustración 40: Diagrama de actividad del caso de uso Marcar incidencia de la aplicación del usuario

VER INFORMACIÓN

Tabla 27: Caso de uso: Ver Información - Aplicación MiTutor		
Descripción	El usuario consultara la información de la aplicación.	
Restricciones	Pre-Condiciones	El usuario deber haber iniciado la aplicación y encontrarse en el menú principal.
	Post-Condiciones	El usuario quedara informado de la información relacionada con la aplicación.
Secuencia de eventos	<b>P.1.</b> El usuario pulsara el botón "Información". <b>P.2.</b> El sistema mostrara la interfaz de la información de la aplicación.	

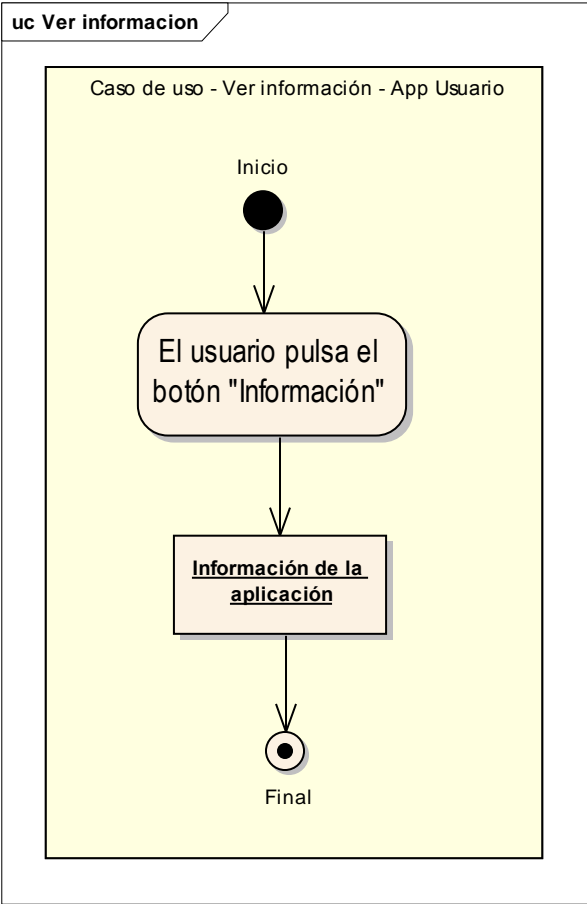


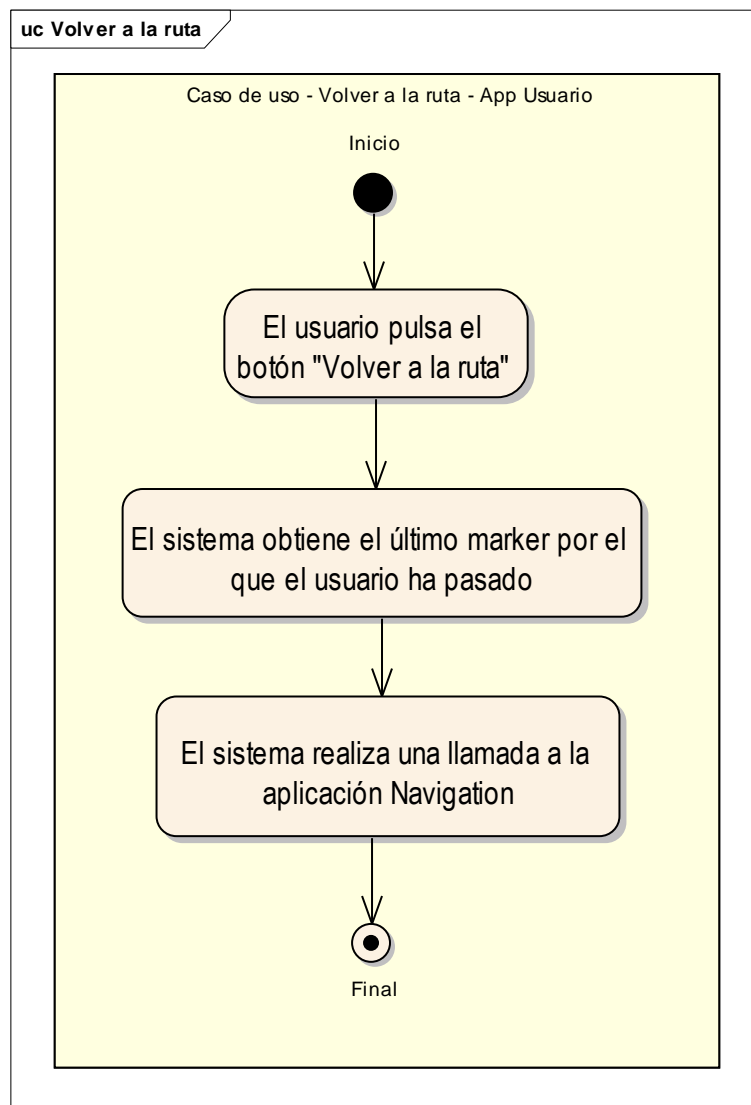
Ilustración 41: Diagrama de actividad del caso de uso Ver información de la aplicación del usuario



## VOLVER A LA RUTA

**Tabla 28: Caso de uso: Volver A La Ruta - Aplicación SoyTutor**

Descripción	El usuario necesita volver a la ruta porque este se ha salido de ella.	
Restricciones	Pre-Condiciones	El usuario deberá haber iniciado una ruta y haberse perdido o salido de la ruta.
	Post-Condiciones	El usuario podrá proseguir con su ruta tras volver a ella.
Secuencia de eventos	<p><b>P.1.</b> El usuario pulsa el botón “Volver a la ruta”.</p> <p><b>P.2.</b> El sistema obtiene los datos del último marker de la ruta por donde el usuario ha pasado.</p> <p><b>P.3.</b> El sistema realiza una llamada a la Aplicación de Navigation de Google Maps con las coordenadas del usuario como origen y las coordenadas del último punto como destino.</p>	



*Ilustración 42: Diagrama de actividad del caso de uso volver a la ruta de la aplicación del usuario*

## Paquetes y clases

### Adapter.java

- *row\_item.xml*: Se trata de una estructura, que representa los elementos de una lista, formada por otros elementos como pueden ser imágenes y texto.
- Método Adapter(Context, ArrayList<T>): Constructor de la clase Adapter, que se encarga de hacer una lista de elementos.
- Método getView(int, View, ViewGroup): Rellena todas las partes de cada elemento de la lista de una forma determinada dependiendo de qué tipo de elemento sea T.
- Método getElement(int): Obtiene el elemento de la posición indicada.

### BaseActivity.java

- Método DesactivarAlarma(): Evita que aparezcan notificaciones.
- Método ActivarAlarma(int): Habilita las notificaciones en la barra superior.
- Método noHayUsuarios(): Indica mediante un mensaje en pantalla que el guía no tiene usuarios registrados.
- Método tutorNoValido(): Indica que las credenciales con las que se ha intentado iniciar sesión, no son válidas.

### ConfiguracionActivity.java

- *Activity\_configuracion.xml*: Estructura de los campos que componen la pantalla de configuración.
- Método CambiarEstadoAlarma(): Método al que se llama al pulsar en el interruptor de activar/desactivar notificaciones, que dependiendo de en qué estado se encuentren se llamará al método ActivarAlarma(int) o DesactivarAlarma().
- Método CambiarTiempoNotif(int): Configura la comprobación de nuevas notificaciones en el intervalo de tiempo que se indique.
- Método onKeyDown(int, KeyEvent): Método que sobrescribe la funcionalidad por defecto al pulsar una tecla. Se ha usado únicamente para el botón *back* (en todas las pantallas vuelve a la pantalla anterior excepto en el menú, que pregunta si se desea salir de la aplicación).

### Guia.java

- Método Guia(int, String, String, String, String): Constructor de la clase Guia.
- Método getIdGuia(): Devuelve el identificador del Guía que lo invoque.
- Método setIdGuia(int): Introduce el identificador en el Guía que lo invoque.
- Método getNombre(): Devuelve el nombre del Guía que lo invoque.
- Método setNombre(String): Introduce el nombre en el Guía que lo invoque.
- Método getCorreo(): Devuelve el email del Guía que lo invoque.
- Método setCorreo(String): Introduce el email en el Guía que lo invoque.
- Método getTelefono(): Devuelve el teléfono del Guía que lo invoque.
- Método setTelefono(String): Introduce el teléfono en el Guía que lo invoque.

- Método `getContrasenna()`: Devuelve la contraseña del Guía que lo invoque.
- Método `setContrasenna(String)`: Introduce la contraseña en el Guía que lo invoque.
- Método `toString()`: Devuelve la información más relevante del Guía que lo invoque.

### **InformacionActivity.java**

- *activity\_informacion.xml*: Pantalla de información acerca de los autores y colaboradores de la aplicación.

### **LoginActivity.java**

- *LoginActivity.xml*: Estructura de la pantalla de inicio de sesión dividida en campos para introducir usuario y contraseña y botones de acceso y registro (para el caso de que sea nuevo en el sistema).
- Método `mensaje(String)`: Muestra por pantalla el mensaje indicado.
- Método `loginstatus(String, String)`: Comprueba que el *username* y *password* se encuentran en la base de datos y devuelve una confirmación (o denegación en caso contrario) de acceso a la aplicación.
- Método `checklogindata(String, String)`: Comprueba que no hay campos vacíos al pulsar en Iniciar Sesión.
- Método `onKeyDown(int, KeyEvent)`: Método que sobrescribe la funcionalidad por defecto al pulsar una tecla. Se ha usado únicamente para el botón *back* (en todas las pantallas vuelve a la pantalla anterior excepto en el menú, que pregunta si se desea salir de la aplicación).

### **MainActivity.java**

- *MainActivity.xml*: Primera pantalla de la aplicación.

### **MapaRutaActivity.java**

- *Activity\_mapa\_ruta.xml*: Pantalla consistente en un mapa con la ruta seleccionada. Si dicha ruta no tuviera tramos no se podrá dibujar y únicamente se mostrará la localización de esta.
- Método `configuracionGPS()`: este método nos da la localización del dispositivo y nos avisa cuando la posición a cambiado (Cada cinco segundos o diez metros).
- Método `onclickBotonMapaIncidencia(View)`: Este método recoge la acción al pulsar el botón “incidencias”.
- Método `onClickBotonPanico(View)`: Este método recoge la acción al pulsar el botón “pánico”.
- Método `onClickBotonVolverRutas(View)`: Este método recoge la acción al pulsar el botón “Volver a la ruta”.
- Método `inicializar_mapa()`: Este método crea el mapa.
- Método `comprobar_gps()`: Comprueba que el GPS está conectado, y si no fuera así advierte de ello para su activación.

- Método `eleccion(int)`: Funcionalidad de las opciones mostradas en el cuadro de diálogo de `comprobar_gps()`.
- Método `mensaje(String)`: Muestra el mensaje indicado por pantalla.
- Método `rutastatus(String)`: Este método sirve para recuperar de la base de datos los tramos de la ruta seleccionada.
- Método `marcar_puntos()`: Este método permite colocar los markers de inicio y de final de cada tramo de la ruta.
- Método `dibujar_lineas()`: Este método dibuja la polilínea que une todos los markers.
- Método `anadir_marker(LatLng, int)`: Este método permite añadir un marker al mapa.

### **MapaIncidenciaActivity.java**

- *Activity\_mapa\_incidencia.xml*: Pantalla consistente en un mapa con un punto (marker) en el lugar donde ha sido marcada la incidencia seleccionada.
- Método `OnClickCancelar(View)`: Método que se llama al pulsar el botón “Cancelar”.
- Método `OnClickBotonIncidencia1(View)`: Método que se llama al pulsar el botón del primer tipo de incidencia(obstaculo).
- Método `OnClickBotonIncidencia2(View)`: Método que se llama al pulsar el botón del segundo tipo de incidencia(inseguridad).
- Método `OnClickBotonIncidencia3(View)`: Método que se llama al pulsar el botón del tercer tipo de incidencia(otros).
- Método `marcarIncidencia(String)`: Introduce la incidencia en base de datos y notifica al Guía.
- Método `comprobar_gps()`: Comprueba que el GPS está conectado, y si no fuera así advierte de ello para su activación.
- Método `eleccion(int)`: Funcionalidad de las opciones mostradas en el cuadro de diálogo de `comprobar_gps()`.
- Método `mensaje(String)`: Muestra el mensaje indicado por pantalla.
- Método `inicializar_mapa()`: Carga el mapa y configure el zoom.
- Método `posicion_actual()`: En caso de tener el GPS activado, detecta la posición actual del dispositivo.
- Método `poner_marker(double,double)`: Dibuja un punto en el mapa.
- Método `notificacionstatus(String, String, String, String, String)`: Marca como vista la incidencia para el tutor que esté usando la aplicación.

### **MenuActivity.java**

- *Activity\_menu.xml*: Estructura de la pantalla menú y sus iconos y botones.
- Método `OnClickBotonRutas(View)`: Método que se llama al pulsar el botón “rutas”.
- Método `OnClickBotonIncidencias(View)`: Método que se llama al pulsar el botón “incidencias”.
- Método `OnClickBotonLlamar(View)`: Método que se llama al pulsar el botón “llamar”.

- Método OnclickBotonConfiguracion(View): Método que se llama al pulsar el botón “configuración”.
- Método OnclickBotonInformacion(View): Método que se llama al pulsar el botón “información”.
- Método OnclickBotonSalir():Método que se llama al pulsar el botón “salir”.

### **NotificacionesService.java**

- Método Notificar(): Muestra una notificación en la barra superior.
- Método notificationsstatus(String): Método que hace la consulta a la base de datos buscando si hay notificaciones que mostrar.

### **RegistroActivity.java**

- *Activity\_registro.xml*: Estructura de los campos para registrar un nuevo tutor.
- Método registrostatus(String, String, String): Hace la inserción de los datos del nuevo tutor en la base de datos.
- Método checkdata(String, String, String): Comprueba que los campos están correctamente llenados.

### **Ruta.java**

- Método Ruta(String, String, String, double, double, double, double, double, int): Constructor de una Ruta
- Método getNombreRuta(): Devuelve el nombre de la Ruta que lo invoque.
- Método setNombreRuta(String): Introduce el nombre en la Ruta que lo invoque.
- Método getInicioRutaNombre(): Devuelve el nombre del origen de la Ruta que lo invoque.
- Método setInicioRutaNombre(String): Introduce el nombre del origen en la Ruta que lo invoque.
- Método getFinRutaNombre(): Devuelve el nombre del destino de la Ruta que lo invoque.
- Método setFinRutaNombre(String): Introduce el nombre del destino en la Ruta que lo invoque.
- Método getInicioRutaLat(): Devuelve la latitud del origen de la Ruta que lo invoque.
- Método setInicioRutaLat(double): Introduce la latitud del origen en la Ruta que lo invoque.
- Método getFinRutaLat(): Devuelve la latitud del destino de la Ruta que lo invoque.
- Método setFinRutaLat(double): Introduce la latitud del destino en la Ruta que lo invoque.
- Método getInicioRutaLng(): Devuelve la longitud del origen de la Ruta que lo invoque.
- Método setInicioRutaLng(double): Introduce la longitud del origen en la Ruta que lo invoque.

- Método `getFinRutaLng()`: Devuelve la longitud del destino de la Ruta que lo invoque.
- Método `setFinRutaLng(double)`: Introduce la longitud del destino en la Ruta que lo invoque.
- Método `getIdRuta()`: Devuelve el identificador de la Ruta que lo invoque.
- Método `setIdRuta(int)`: Introduce el identificador en la Ruta que lo invoque.
- Método `getValoracion()`: Devuelve la valoración de la Ruta que lo invoque.
- Método `setValoracion(double)`: Introduce la valoración en la Ruta que lo invoque.
- Método `toString()`: Devuelve la información más relevante de la Incidencia que lo invoque.

### **RutasActivity.java**

- *Activity\_rutas.xml*: Estructura de la lista de rutas.
- Método `getRutasUsuario(String idGuia)`: Hace la llamada a la base de datos y devuelve las rutas del usuario.

### **TemplateActivity.java**

- Método `onCreateOptionsMenu(Menu)`: Crea el menú común a todas las pantallas.
- Método `onOptionsItemSelected(MenuItem)`: Establece la acción al pulsar en una opción del menú.

### **Tramo.java**

- Método `Tramo(int,double,double,double,double,double, double, int)`: Constructor de un Tramo.
- Método `getIdTramo()`: Devuelve el identificador del Tramo que lo invoque.
- Método `setValoracion(double)`: Introduce el identificador en el Tramo que lo invoque.
- Método `getInicioLat()`: Devuelve la latitud del inicio del Tramo que lo invoque.
- Método `getInicioLng()`: Devuelve la longitud del inicio del Tramo que lo invoque.
- Método `getFinalLat()`: Devuelve la latitud del fin del Tramo que lo invoque.
- Método `getFinalLng()`: Devuelve la longitud del fin del Tramo que lo invoque.
- Método `getValoracionDer()`: Devuelve la valoración de la acera derecha del Tramo que lo invoque.
- Método `getValoracionIzq()`: Devuelve la valoración de la acera izquierda del Tramo que lo invoque.
- Método `getOrden()`: Devuelve el orden del Tramo que lo invoque.

### **Usuario.java**

- Método `Usuario(int, String, String, String, String, String, String, String, String, String, Guia)`: Constructor de un usuario.
- Método `getIdUsuario()`: Devuelve el identificador del Usuario que lo invoque.

- Método setIdUsuario(int): Introduce el identificador en el Usuario que lo invoque.
- Método getNombre(): Devuelve la nombre del Usuario que lo invoque.
- Método setNombre(String): Introduce el nombre en el Usuario que lo invoque.
- Método getApellido1(): Devuelve el primer apellido del Usuario que lo invoque.
- Método setApellido1(String): Introduce el primer apellido en el Usuario que lo invoque.
- Método getApellido2(): Devuelve el segundo apellido Usuario que lo invoque.
- Método setApellido2(String): Introduce el segundo apellido en el Usuario que lo invoque.
- Método getDireccion(): Devuelve la dirección del Usuario que lo invoque.
- Método setDireccion(String): Introduce la dirección en el Usuario que lo invoque.
- Método getLocalidad(): Devuelve la localidad del Usuario que lo invoque.
- Método setLocalidad(String): Introduce la localidad en el Usuario que lo invoque.
- Método getProvincia(): Devuelve la provincia del Usuario que lo invoque.
- Método setProvincia(String): Introduce la provincia en el Usuario que lo invoque.
- Método getTelefono1(): Devuelve el teléfono principal del Usuario que lo invoque.
- Método setTelefono1(String): Introduce el teléfono principal en el Usuario que lo invoque.
- Método getTelefono2(): Devuelve el teléfono secundario del Usuario que lo invoque.
- Método setTelefono2(String): Introduce el teléfono secundario en el Usuario que lo invoque.
- Método getEdad(): Devuelve la edad del Usuario que lo invoque.
- Método setEdad(String): Introduce la edad en el Usuario que lo invoque.
- Método getGuia(): Devuelve el Guia del Usuario que lo invoque.
- Método setGuia(Guia): Introduce el Guia en el Usuario que lo invoque.
- Método toString(): Devuelve la información más relevante del Usuario que lo invoque.

### **UsuariosActivity.java**

- *Activity\_usuarios.xml*: Estructura de la lista de usuarios.
- Método mensaje(String): Método mensaje(String): Muestra por pantalla el mensaje indicado.
- Método usuariosstatus(String): Hace una llamada a la base de datos y devuelve los usuarios del guía que haya iniciado sesión en la aplicación.
- Método checklogindata(String): Comprueba que los campos están correctamente rellenos.

## **Paquetes utilizados en el desarrollo de la aplicación**

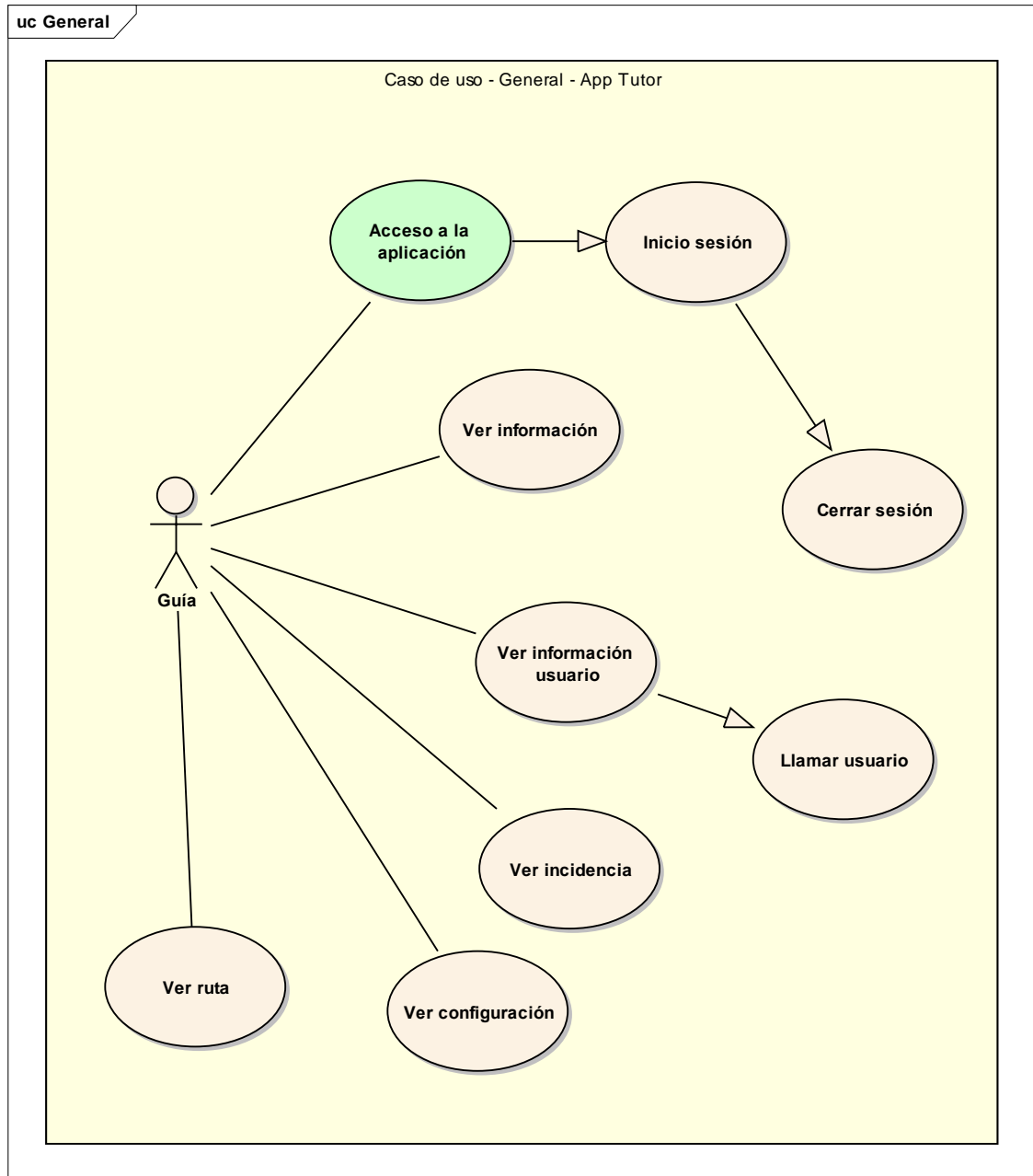
- **Android.widget:** El paquete contiene elementos de la interfaz de usuario para usar en la pantalla de la aplicación.
- **Android.view:** Este paquete proporciona clases básicas de interfaz de usuario que se encargan del diseño de la pantalla y la interacción con el usuario.
- **Android.content:** Este paquete contiene clases para el acceso y la publicación de datos en el dispositivo.
- **Android.app:** Este paquete contiene clases de alto nivel que encapsulan el modelo general de la aplicación Android.
- **Android.os:** Este paquete proporciona servicios básicos del sistema operativo, el paso de mensajes y la comunicación entre procesos en el dispositivo.
- **Android.graphics:** Este paquete proporciona herramientas de gráficos de bajo nivel, como lienzos, filtros de color, puntos y rectángulos que permiten dibujar en la pantalla directamente.
- **Android.location:** Este paquete contiene las clases del framework API que definen los servicios basados en la localización de Android.
- **Android.net:** Este paquete proporciona las clases para el acceso a la red.
- **Android.provider:** Este paquete proporciona clases de conveniencia para acceder a los proveedores de contenidos suministrados por Android.
- **Java.util:** Este paquete contiene el marco de las colecciones y clases de colecciones tradicionales.
- **Org.apache.http:** Este paquete se ocupa de las cosas fundamentales que requiere utilizar el protocolo HTTP.
- **Com.google.android.gms.maps:** Este paquete contiene los mapas de Google de la API de Android.



## 4.6 Arquitectura de la Aplicación del guía (SoyTutor)

El propósito de esta sección es explicar las diferentes funcionalidades de la aplicación del guía a través de casos de uso y diagramas de actividades. Además se analizan las diferentes clases utilizadas en esta aplicación ayudándonos de una breve explicación para cada una de ellas.

### Casos de Uso



*Ilustración 43: Diagrama general de casos de uso de la aplicación del guía*

CERRAR SESIÓN

Tabla 29: Caso de uso: Cerrar Sesión - Aplicación SoyTutor		
Descripción	El usuario querrá cerrar la sesión de la aplicación.	
Restricciones	Pre-Condiciones	El guía debe haber iniciado la aplicación y encontrarse en el menú principal.
	Post-Condiciones	El guía habrá cerrado la sesión de la aplicación.
Secuencia de eventos	<p><b>P.1.</b> El usuario pulsara el botón "Salir" o bien el botón "atrás" del terminal.</p> <p><b>P.2.</b> El sistema mandara un mensaje al usuario para confirmar la acción.</p> <p><b>P.2.1.</b> El usuario pulsa el botón "Si".</p> <p><b>P.2.1.1.</b> El sistema cerrara la aplicación.</p> <p><b>P.2.2.</b> El usuario pulsa el botón "No".</p> <p><b>P.2.2.1.</b> El sistema regresara al menú principal.</p>	

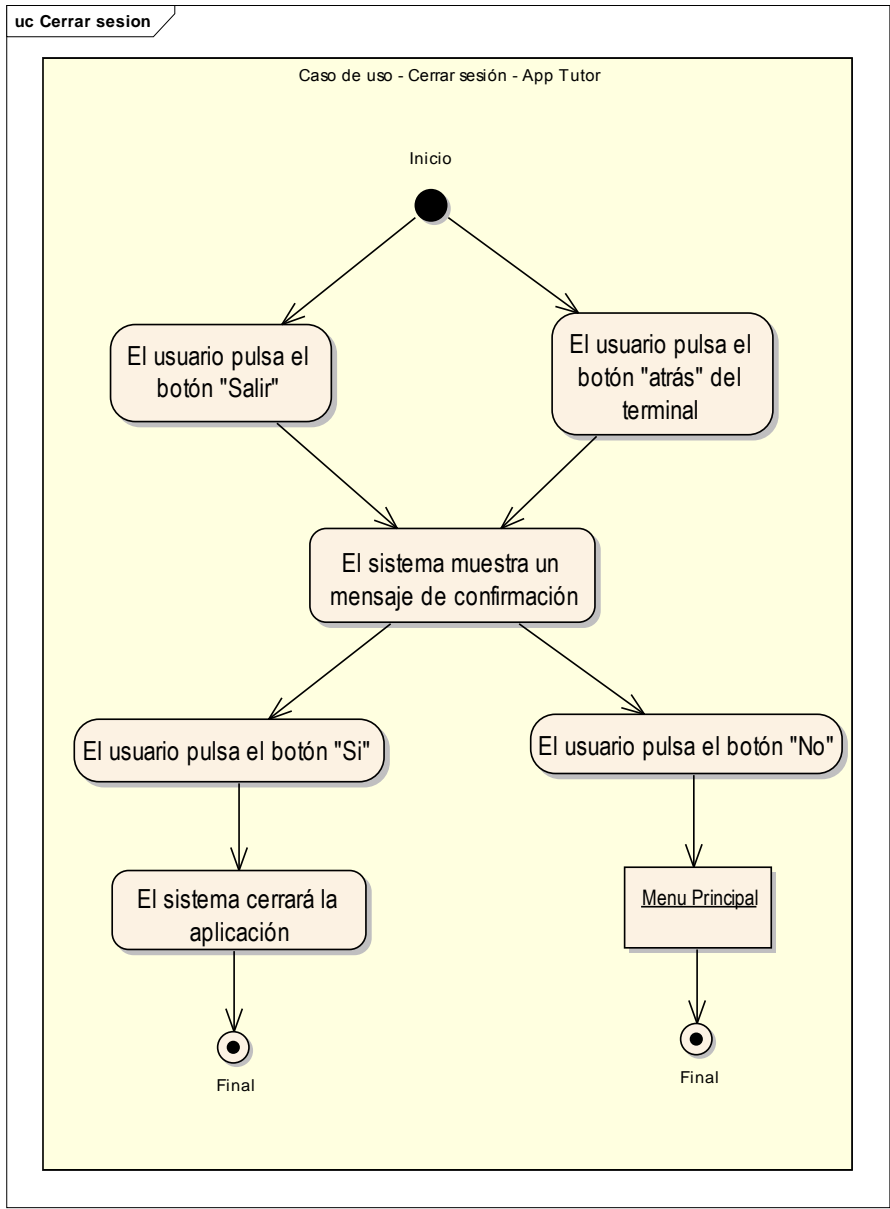


Ilustración 44: Diagrama de actividad del caso de uso Cerrar sesión de la aplicación del guía

## INICIO SESIÓN

Tabla 30: Caso de uso: Inicio Sesión - Aplicación SoyTutor		
Descripción	Acceso a la aplicación SoyTutor por parte del guía.	
Restricciones	Pre-Condiciones	El guía debe estar registrado en la base de datos.
	Post-Condiciones	Quedara iniciada la sesión del guía para el posterior uso de las funcionalidades de la aplicación.
Secuencia de eventos	<p><b>P.1.</b> El sistema busca si los datos de acceso están ya guardados en la memoria del teléfono.</p> <p><b>P.1.1.</b> El sistema encuentra al guía en los datos de la aplicación.</p> <p><b>P.1.1.1.</b> El sistema nos lleva al menú principal de la aplicación.</p> <p><b>P.1.2.</b> El sistema no encuentra al guía en los datos de la aplicación.</p> <p><b>P.1.2.1.</b> El sistema nos lleva a la pantalla de inicio de sesión de la aplicación.</p> <p><b>P.1.2.2.</b> El usuario introducirá los datos requeridos por la aplicación para el inicio de sesión (Correo y Contraseña).</p> <p><b>P.1.2.3.</b> El usuario pulsara el botón "Entrar".</p> <p><b>P.1.2.3.1.</b> El sistema busca los datos de acceso en a base de datos.</p> <p><b>P.1.2.3.1.1.</b> El sistema encuentra al guía en base de datos.</p> <p><b>P.1.2.3.1.1.1.</b> El sistema guardara la información del guía en la memoria del teléfono para futuros accesos.</p> <p><b>P.1.2.3.1.1.2.</b> El sistema nos llevara al menú principal de la aplicación.</p> <p><b>P.1.2.3.1.2.</b> El sistema no encuentra al guía en la base de datos.</p> <p><b>P.1.2.3.1.2.1.</b> La aplicación nos informara "Los datos de acceso no son correctos".</p>	

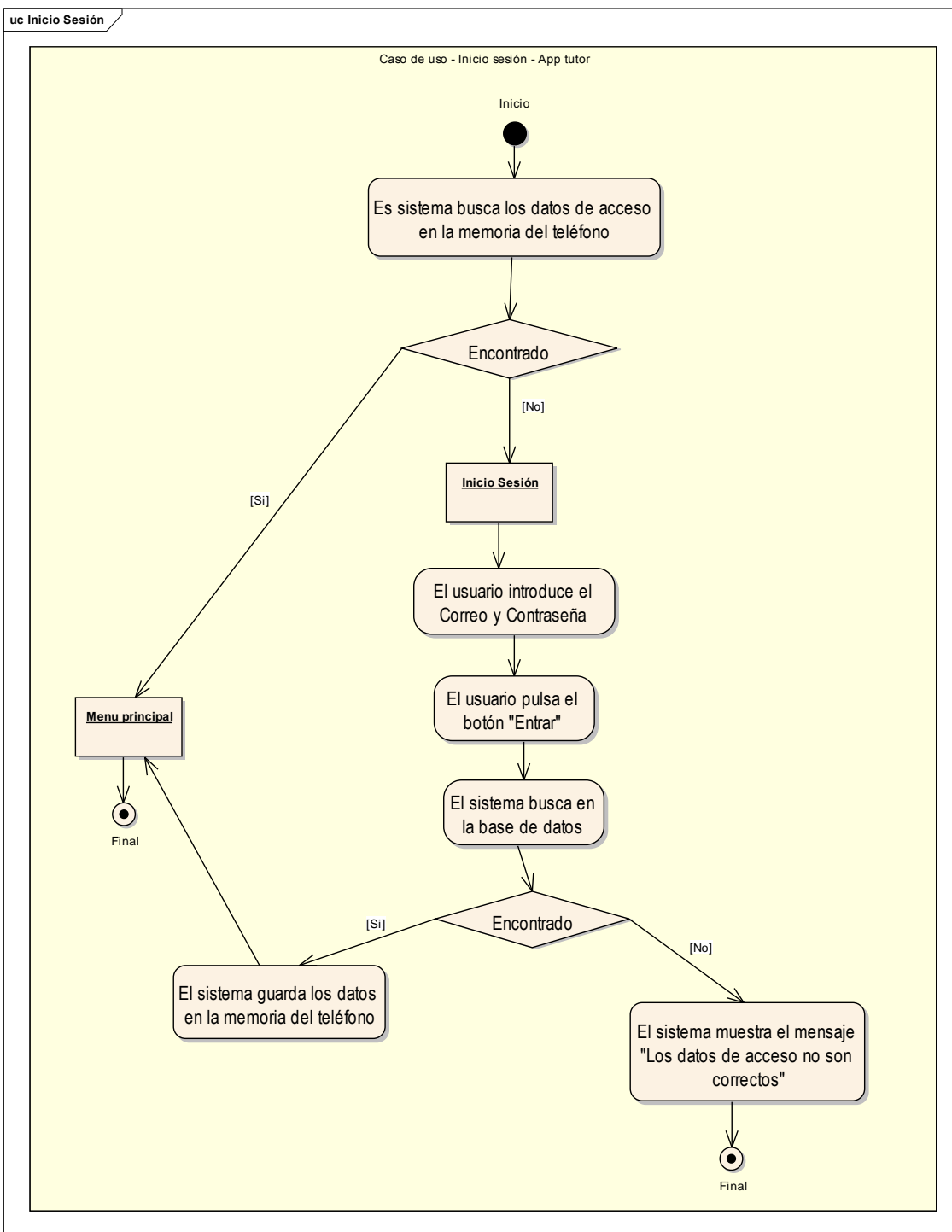


Ilustración 45: Diagrama de actividad del caso de uso Inicio sesión de la aplicación del guía

LLAMAR USUARIO

Tabla 31: Caso de uso: Llamar Usuario - Aplicación SoyTutor		
Descripción	El guía podrá ponerse en contacto sus usuarios telefónicamente.	
Restricciones	Pre-Condiciones	El guía deberá accedido a la aplicación y haber accedido a la información del usuario, según el caso de uso "ver información usuario".
	Post-Condiciones	El guía se habrá comunicado con su usuario.
Secuencia de eventos	<b>P.1.</b> El usuario pulsara el botón "Llamar" que se encuentra en el "Dialog" de la información del usuario. <b>P.2.</b> El sistema consulta el número de teléfono del usuario. <b>P.3.</b> El sistema realiza la llamada telefónica al usuario.	

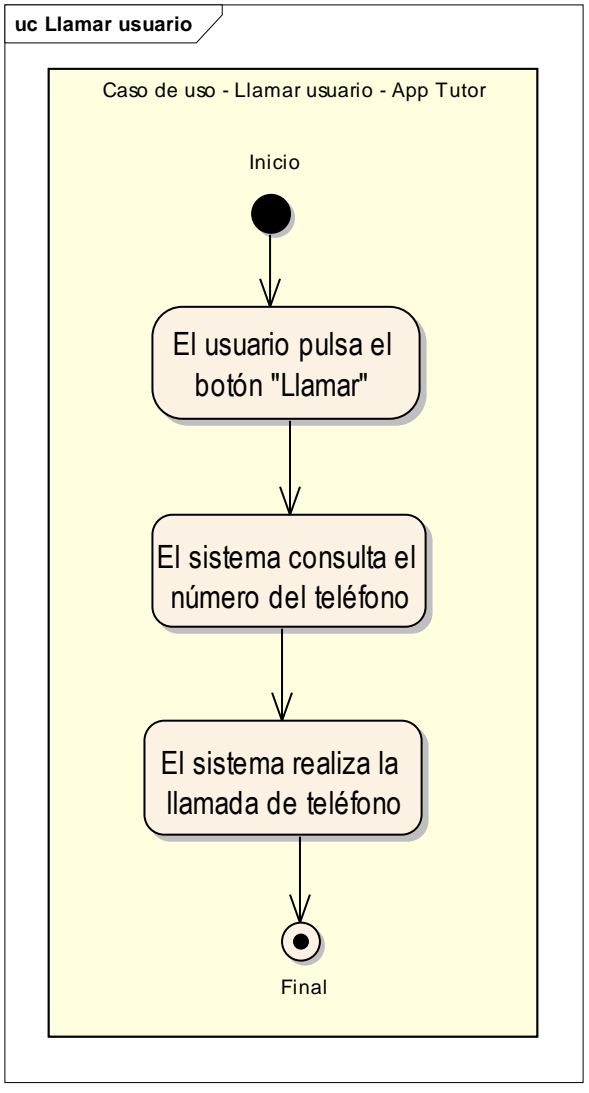


Ilustración 46: Diagrama de actividad del caso de uso Llamar usuario de la aplicación del guía

VER CONFIGURACIÓN

Tabla 32: Caso de uso: Ver Configuración - Aplicación SoyTutor		
Descripción	El guía consultara y podrá modificar su configuración de la aplicación.	
Restricciones	Pre-Condiciones	El guía deber haber iniciado la aplicación y encontrarse en el menú principal.
	Post-Condiciones	El guía habrá comprobado su configuración de la aplicación.
Secuencia de eventos	<b>P.1.</b> El usuario pulsara el botón "Configuración". <b>P.2.</b> El sistema mostrara la interfaz de configuración de la aplicación.	

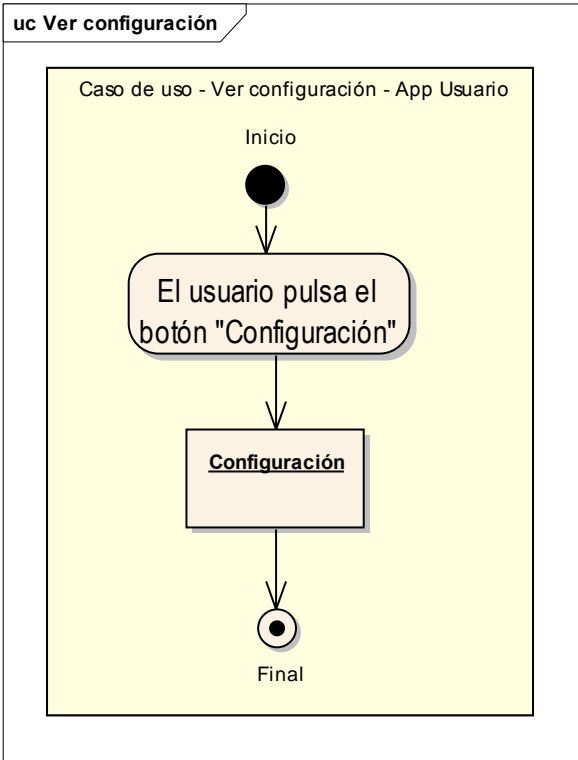


Ilustración 47: Diagrama de actividad del caso de uso Ver configuración de la aplicación del guía

## VER INCIDENCIA

Tabla 33: Caso de uso: Ver Incidencia - Aplicación SoyTutor		
Descripción	El guía podrá consultar las incidencias de sus usuarios.	
Restricciones	Pre-Condiciones	El guía debe haber iniciado la aplicación y encontrarse en el menú principal.
	Post-Condiciones	El guía habrá visualizado la incidencia deseada.
Secuencia de eventos	<p><b>P.1.</b> El usuario pulsara el botón "Incidencias".</p> <p><b>P.2.</b> El sistema buscara las incidencias asociadas a los usuarios del guía en la base de datos.</p> <p><b>P.2.1.</b> El sistema encuentra incidencias</p> <p><b>P.2.1.1.</b> El sistema mostrara un listado con todas las incidencias.</p> <p><b>P.2.1.2.</b> El usuario seleccionara la incidencia que desea visualizar.</p> <p><b>P.2.1.3.</b> El sistema consulta con la base de datos los datos relacionados con la incidencia seleccionada.</p> <p><b>P.2.1.4.</b> El sistema mostrara en un mapa la incidencia seleccionada.</p> <p><b>P.2.2.</b> El sistema no encuentra rutas.</p> <p><b>P.2.2.1.</b> El sistema muestra el mensaje "Ninguno de sus usuarios ha marcado incidencias".</p>	

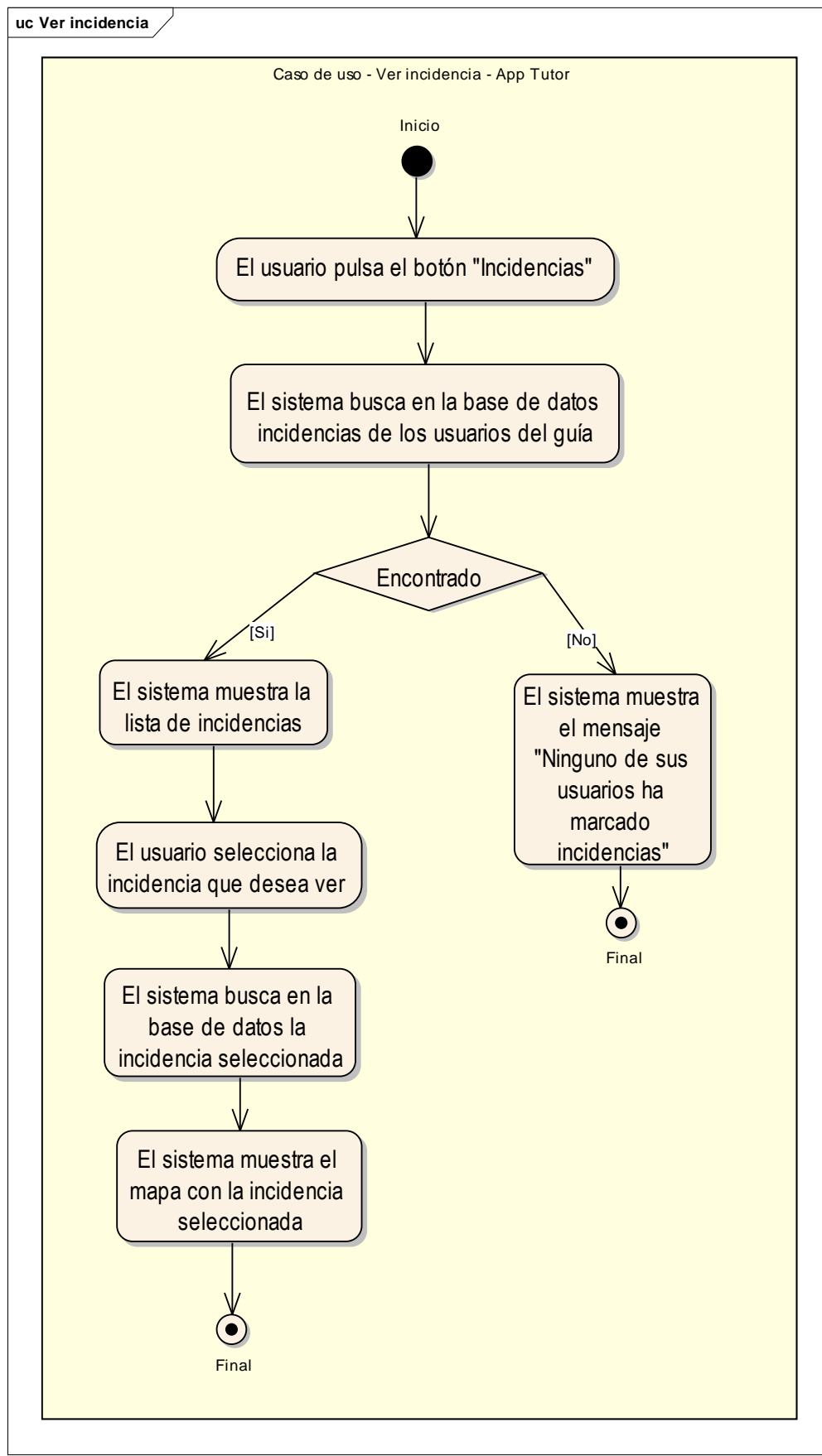


Ilustración 48: Diagrama de actividad del caso de uso Ver incidencia de la aplicación del guía



## VER INFORMACIÓN USUARIO

Tabla 34: Caso de uso: Ver Información Usuario - Aplicación SoyTutor		
Descripción	El guía podrá consultar la información de sus usuarios.	
Restricciones	Pre-Condiciones	El guía debe haber iniciado la aplicación y encontrarse en el menú principal.
	Post-Condiciones	El guía habrá consultado la información del usuario.
Secuencia de eventos	<p><b>P.1.</b> El usuario pulsara el botón "Usuarios".</p> <p><b>P.2.</b> El sistema busca en la base de datos los usuarios pertenecientes al guía.</p> <p><b>P.2.1.</b> El sistema encuentra usuarios a su cargo.</p> <p><b>P.2.1.1.</b> El sistema muestra una lista con todos los usuarios.</p> <p><b>P.2.1.2.</b> El usuario pulsa sobre el "usuario" que quiera ver la información.</p> <p><b>P.2.1.3.</b> El sistema consulta los datos de ese usuario.</p> <p><b>P.2.1.4.</b> El sistema abre un "Dialog" mostrando la información del usuario.</p> <p><b>P.2.2.</b> El sistema no encuentra usuarios a su cargo.</p> <p><b>P.2.2.1.</b> El sistema le mostrara el mensaje "No tiene usuarios a su cargo".</p>	

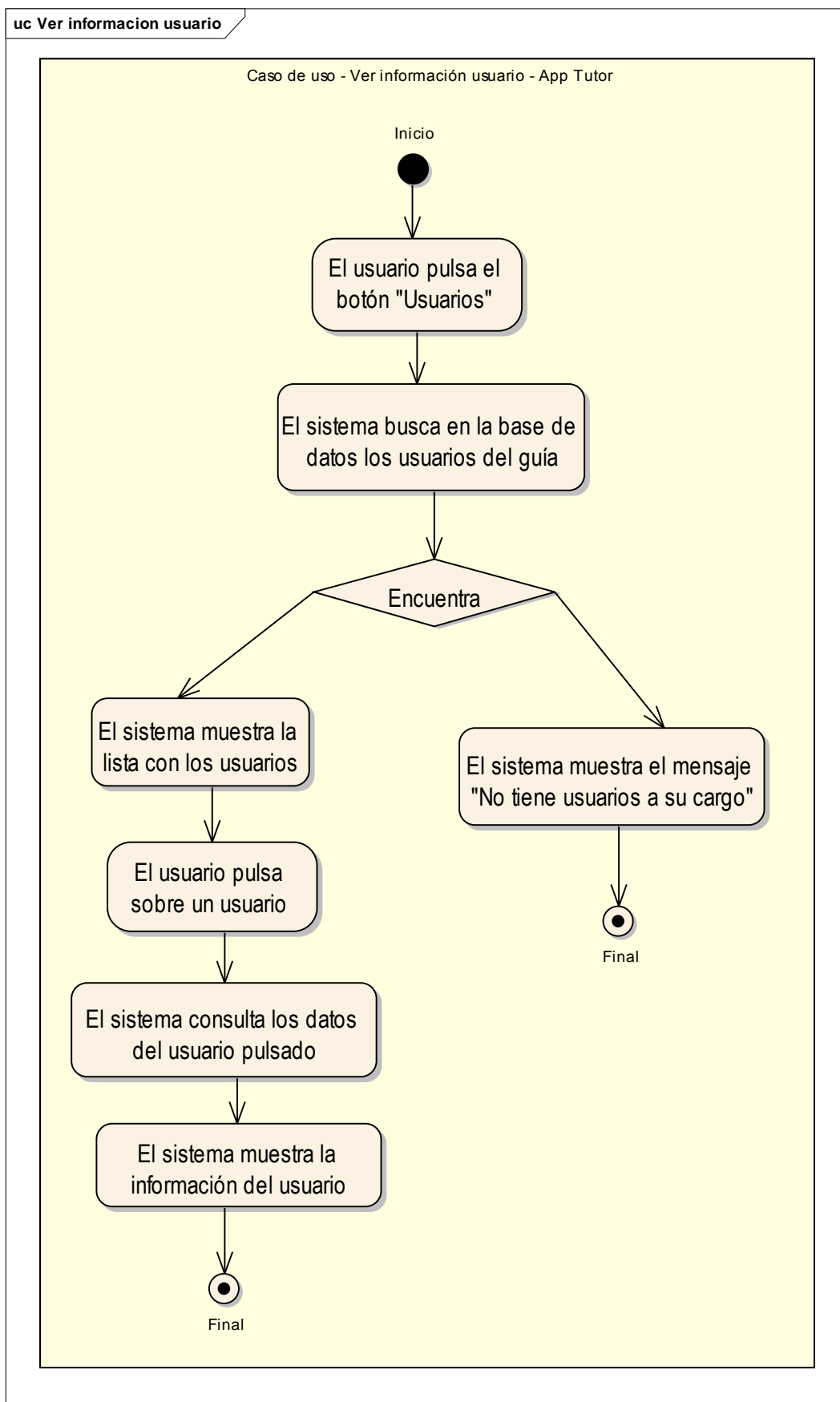


Ilustración 49: Diagrama de actividad del caso de uso Ver información del usuario de la aplicación del guía

VER INFORMACIÓN

Tabla 35: Caso de uso: Ver Información - Aplicación SoyTutor		
Descripción	El usuario consultara la información de la aplicación.	
Restricciones	Pre-Condiciones	El guía deber haber iniciado la aplicación y encontrarse en el menú principal.
	Post-Condiciones	El guía quedara informado de la información relacionada con la aplicación.
Secuencia de eventos	<b>P.1.</b> El usuario pulsara el botón "Información". <b>P.2.</b> El sistema mostrara la interfaz de la información de la aplicación.	

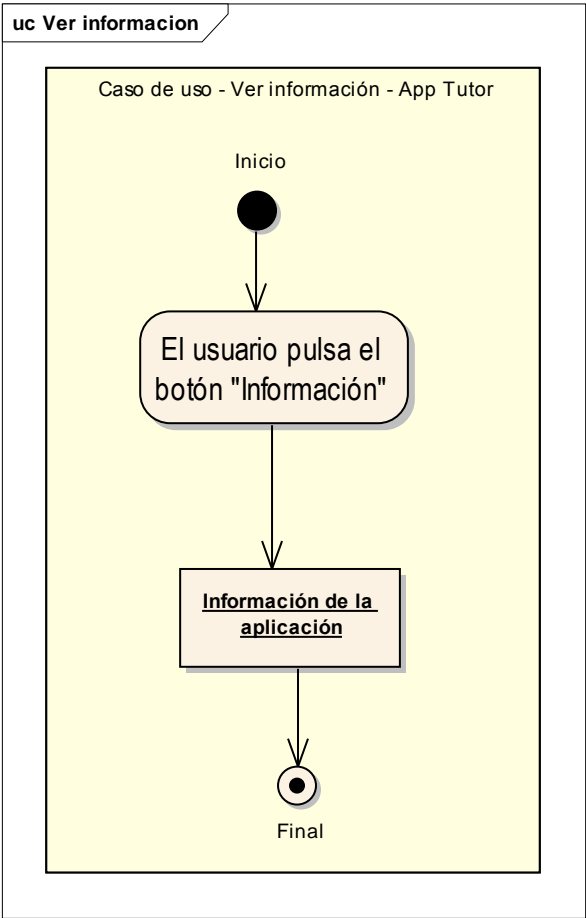


Ilustración 50: Diagrama de actividad del caso de uso Ver información de la aplicación del guía

## VER RUTA

Tabla 36: Caso de uso: Ver Ruta - Aplicación SoyTutor		
Descripción	El guía podrá consultar las rutas de sus usuarios.	
Restricciones	Pre-Condiciones	El guía debe haber iniciado la aplicación y encontrarse en el menú principal.
	Post-Condiciones	El guía habrá visualizado la ruta deseada.
Secuencia de eventos	<p><b>P.1.</b> El usuario pulsara el botón "Rutas usuario".</p> <p><b>P.2.</b> El sistema buscara las rutas asociadas a los usuarios del guía en la base de datos.</p> <p><b>P.2.1.</b> El sistema encuentra rutas</p> <p><b>P.2.1.1.</b> El sistema mostrara un listado con todas las rutas.</p> <p><b>P.2.1.2.</b> El usuario seleccionara la ruta que desea visualizar.</p> <p><b>P.2.1.3.</b> El sistema consulta con la base de datos los tramos relacionados con la ruta seleccionada.</p> <p><b>P.2.1.3.1.</b> La ruta esta valorada</p> <p><b>P.2.1.3.1.1.</b> El sistema mostrara la ruta visualizando la información de los tramos, inicio y fin de la ruta.</p> <p><b>P.2.1.3.2.</b> La ruta no esta valorada</p> <p><b>P.2.1.3.2.1.</b> El sistema mostrara la ruta visualizando la información de inicio y final de la ruta.</p> <p><b>P.2.1.3.2.2.</b> El sistema muestra el mensaje “La ruta seleccionada no tiene tramos definidos”</p> <p><b>P.2.2.</b> El sistema no encuentra rutas.</p> <p><b>P.2.2.1.</b> El sistema muestra el mensaje "Ninguno de sus usuarios tiene rutas definidas".</p>	

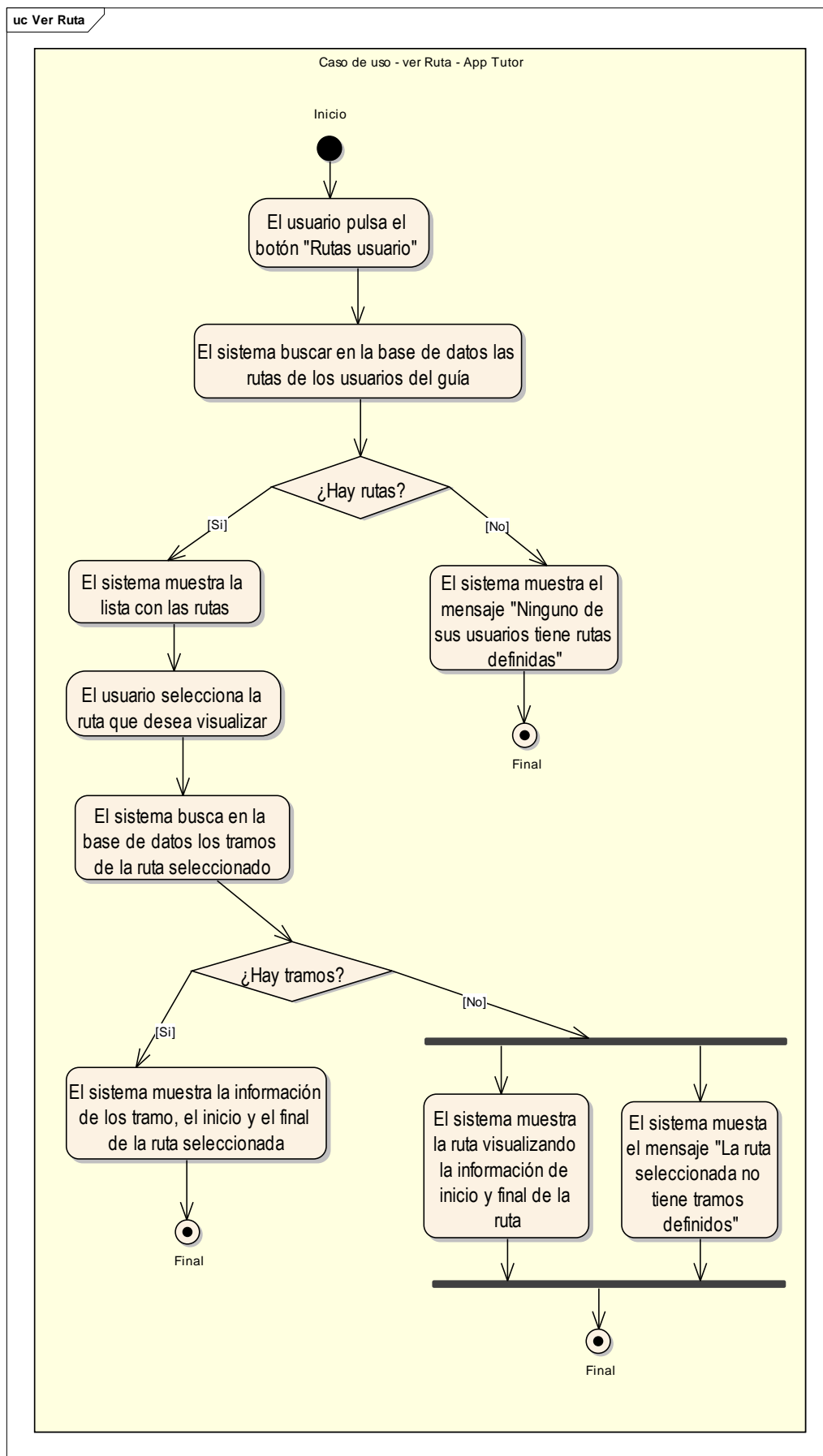


Ilustración 51: Diagrama de actividad del caso de uso Ver ruta de la aplicación del guía

## Paquetes y clases

### Adapter.java

- *row\_item.xml*: Se trata de una estructura, que representa los elementos de una lista, formada por otros elementos como pueden ser imágenes y texto.
- Método `Adapter(Context, ArrayList<T>)`: Constructor de la clase Adapter, que se encarga de hacer una lista de elementos.
- Método `getView(int, View, ViewGroup)`: Rellena todas las partes de cada elemento de la lista de una forma determinada dependiendo de qué tipo de elemento sea T.
- Método `getElement(int)`: Obtiene el elemento de la posición indicada.

### AdapterSimple.java

- *row\_item\_simple.xml*: Se trata de una estructura, que representa los elementos de una lista, formada por otros elementos como pueden ser imágenes y texto.
- Método `Adapter(Context, ArrayList<T>)`: Constructor de la clase Adapter, que se encarga de hacer una lista de elementos.
- Método `getView(int, View, ViewGroup)`: Rellena todas las partes de cada elemento de la lista de una forma determinada dependiendo de qué tipo de elemento sea T.
- Método `getElement(int)`: Obtiene el elemento de la posición indicada.

### BaseActivity.java

- Método `DesactivarAlarma()`: Evita que aparezcan notificaciones.
- Método `ActivarAlarma(int)`: Habilita las notificaciones en la barra superior, configurando la comprobación de nuevas notificaciones en el intervalo de tiempo que se indique.
- Método `noHayUsuarios()`: Indica mediante un mensaje en pantalla que el guía no tiene usuarios registrados.
- Método `noHayIncidencias()`: Indica mediante un mensaje en pantalla que los usuarios que tiene a cargo el guía no han marcado ninguna incidencia.
- Método `noHayRutas()`: Indica mediante un mensaje en pantalla que los usuarios que tiene a cargo el Guía no tienen rutas configuradas.
- Método `tutorNoValido()`: Indica que las credenciales con las que se ha intentado iniciar sesión, no son válidas.
- Método `rutaSinTramos()`: Indica que la ruta que ha seleccionado, no está dividida en tramos.
- Método `errorRegistro()`: Indica que ha habido un error al registrar un tutor nuevo.

## ConfiguracionActivity.java

- *Activity\_configuracion.xml*: Estructura de los campos que componen la pantalla de configuración.
- Método `CambiarEstadoAlarma()`: Método al que se llama al pulsar en el interruptor de activar/desactivar notificaciones, que dependiendo de en qué estado se encuentren se llamará al método `ActivarAlarma(int)` o `DesactivarAlarma()`.
- Método `CambiarTiempoNotif(int)`: Configura la comprobación de nuevas notificaciones en el intervalo de tiempo que se indique.
- Método `onKeyDown(int, KeyEvent)`: Método que sobrescribe la funcionalidad por defecto al pulsar una tecla. Se ha usado únicamente para el botón *back* (en todas las pantallas vuelve a la pantalla anterior excepto en el menú, que pregunta si se desea salir de la aplicación).

## Guia.java

- Método `Guia(int, String, String, String, String)`: Constructor de la clase Guía.
- Método `getIdGuia()`: Devuelve el identificador del Guía que lo invoque.
- Método `setIdGuia(int)`: Introduce el identificador en el Guía que lo invoque.
- Método `getNombre()`: Devuelve el nombre del Guía que lo invoque.
- Método `setNombre(String)`: Introduce el nombre en el Guía que lo invoque.
- Método `getCorreo()`: Devuelve el email del Guía que lo invoque.
- Método `setCorreo(String)`: Introduce el email en el Guía que lo invoque.
- Método `getTelefono()`: Devuelve el teléfono del Guía que lo invoque.
- Método `setTelefono(String)`: Introduce el teléfono en el Guía que lo invoque.
- Método `getContrasenna()`: Devuelve la contraseña del Guía que lo invoque.
- Método `setContrasenna(String)`: Introduce la contraseña en el Guía que lo invoque.
- Método `toString()`: Devuelve la información más relevante del Guía que lo invoque.

## Incidencia.java

- Método `Incidencia(String, int, double, double, int)`.
- Método `getIdIncidencia()`: Devuelve el identificador de la Incidencia que lo invoque.
- Método `setIdIncidencia(int)`: Introduce el identificador en la Incidencia que lo invoque.
- Método `getNombre()`: Devuelve el nombre de la Incidencia que lo invoque.
- Método `setNombre(String)`: Introduce el nombre en la Incidencia que lo invoque.
- Método `getLatitud()`: Devuelve la latitud de la Incidencia que lo invoque.
- Método `setLatitud(double)`: Introduce la latitud en la Incidencia que lo invoque.
- Método `getLongitud()`: Devuelve la longitud de la Incidencia que lo invoque.
- Método `setLongitud(double)`: Introduce la longitud en la Incidencia que lo invoque.

- Método `getVista()`: Devuelve el identificador de vista o no de la Incidencia que lo invoque.
- Método `setVista(int)`: Introduce el identificador de vista o no en la Incidencia que lo invoque.
- Método `toString()`: Devuelve la información más relevante de la Incidencia que lo invoque.

### **IncenciasActivity.java**

- *Activity\_incencias.xml*: Estructura de la lista de incidencias.
- Método `getIncenciasGuia(String)`: Hace la llamada a la base de datos para consultar las incidencias de los usuarios del Guía indicado.

### **InformacionActivity.java**

- *activity\_informacion.xml*: Pantalla de información acerca de los autores y colaboradores de la aplicación.

### **LoginActivity.java**

- *LoginActivity.xml*: Estructura de la pantalla de inicio de sesión dividida en campos para introducir usuario y contraseña y botones de acceso y registro (para el caso de que sea nuevo en el sistema).
- Método `mensaje(String)`: Muestra por pantalla el mensaje indicado.
- Método `loginstatus(String, String)`: Comprueba que el *username* y *password* se encuentran en la base de datos y devuelve una confirmación (o denegación en caso contrario) de acceso a la aplicación.
- Método `checklogindata(String, String)`: Comprueba que no hay campos vacíos al pulsar en Iniciar Sesión.
- Método `onKeyDown(int, KeyEvent)`: Método que sobrescribe la funcionalidad por defecto al pulsar una tecla. Se ha usado únicamente para el botón *back* (en todas las pantallas vuelve a la pantalla anterior excepto en el menú, que pregunta si se desea salir de la aplicación).

### **MainActivity.java**

- *MainActivity.xml*: Primera pantalla de la aplicación.

### **MapaIncidenciaActivity.java**

- *Activity\_mapa\_incidencia.xml*: Estructura que se compone de un mapa y los botones para que un usuario pueda reportar una incidencia.
- Método `inicializar_mapa()`: Inicializa el mapa y el zoom.
- Método `colocar_marker(LatLng)`: Coloca un marcador en la posición indicada.
- Método `setVistaIncidencia(String)`: Marca como vista la incidencia que se está visualizando.



- Método `onKeyDown(int, KeyEvent)`: Método `onKeyDown(int, KeyEvent)`: Método que sobrescribe la funcionalidad por defecto al pulsar una tecla. Se ha usado únicamente para el botón *back* (en todas las pantallas vuelve a la pantalla anterior excepto en el menú, que pregunta si se desea salir de la aplicación).

### **MapaRutaActivity.java**

- *Activity\_mapa\_ruta.xml*: Estructura que se compone de un mapa y los botones para que un usuario pueda seguir una ruta configurada por su guía y una serie de opciones más.
- Método `configuracionGPS()`: Método para la localización del dispositivo.
- Método `inicializar_mapa()`: Inicializa el mapa y el zoom.
- Método `mensaje(String)`: Muestra por pantalla el mensaje indicado.
- Método `rutastatus(String)`: Hace la llamada a la base de datos para obtener la ruta deseada sus argumentos.
- Método `marcar_puntos()`: Pone los marcadores al principio y final de cada tramo.
- Método `dibujar_lineas()`: Une los marcadores de principio y fin de cada tramo con líneas.
- Método `anadir_marker(LatLng, int)`: Pone un marcador en la posición indicada.

### **MenuActivity.java**

- *Activity\_menu.xml*: Estructura del menú de todas las funcionalidades generales de la aplicación.
- Método `OnClickBotonUsuarios(View)`: Método que se llama al pulsar sobre “usuarios”.
- Método `OnClickBotonIncidencias(View)`: Método que se llama al pulsar sobre “incidencias”.
- Método `OnClickBotonRutas(View)`: Método que se llama al pulsar sobre “rutas”.
- Método `OnClickBotonInformacion(View)`: Método que se llama al pulsar sobre “información”.
- Método `OnClickBotonSalir(View)`: Método que se llama al pulsar sobre “salir”.
- Método `OnClickBotonConfiguracion(View)`: Método que se llama al pulsar sobre “configuración”.
- Método `onKeyDown(int, KeyEvent)`: Método que sobrescribe la funcionalidad por defecto al pulsar una tecla. Se ha usado únicamente para el botón *back* (en todas las pantallas vuelve a la pantalla anterior excepto en el menú, que pregunta si se desea salir de la aplicación).

### **NotificacionesService.java**

- Método `Notificar()`: Muestra una notificación en la barra superior.
- Método `notificationsstatus(String)`: Método que hace la consulta a la base de datos buscando si hay notificaciones que mostrar.

## **RegistroActivity.java**

- *Activity\_registro.xml*: Estructura de los campos para registrar un nuevo tutor.
- Método `registrostatus(String, String, String)`: Hace la inserción de los datos del nuevo tutor en la base de datos.
- Método `checkdata(String, String, String)`: Comprueba que los campos están correctamente llenados.

## **Ruta.java**

- Método `Ruta(String, String, String, double, double, double, double, double, int)`: Constructor de una Ruta.
- Método `getNombreRuta()`: Devuelve el nombre de la Ruta que lo invoque.
- Método `setNombreRuta(String)`: Introduce el nombre en la Ruta que lo invoque.
- Método `getInicioRutaNombre()`: Devuelve el nombre del origen de la Ruta que lo invoque.
- Método `setInicioRutaNombre(String)`: Introduce el nombre del origen en la Ruta que lo invoque.
- Método `getFinRutaNombre()`: Devuelve el nombre del destino de la Ruta que lo invoque.
- Método `setFinRutaNombre(String)`: Introduce el nombre del destino en la Ruta que lo invoque.
- Método `getInicioRutaLat()`: Devuelve la latitud del origen de la Ruta que lo invoque.
- Método `setInicioRutaLat(double)`: Introduce la latitud del origen en la Ruta que lo invoque.
- Método `getFinRutaLat()`: Devuelve la latitud del destino de la Ruta que lo invoque.
- Método `setFinRutaLat(double)`: Introduce la latitud del destino en la Ruta que lo invoque.
- Método `getInicioRutaLng()`: Devuelve la longitud del origen de la Ruta que lo invoque.
- Método `setInicioRutaLng(double)`: Introduce la longitud del origen en la Ruta que lo invoque.
- Método `getFinRutaLng()`: Devuelve la longitud del destino de la Ruta que lo invoque.
- Método `setFinRutaLng(double)`: Introduce la longitud del destino en la Ruta que lo invoque.
- Método `getIdRuta()`: Devuelve el identificador de la Ruta que lo invoque.
- Método `setIdRuta(int)`: Introduce el identificador en la Ruta que lo invoque.
- Método `getValoracion()`: Devuelve la valoración de la Ruta que lo invoque.
- Método `setValoracion(double)`: Introduce la valoración en la Ruta que lo invoque.
- Método `toString()`: Devuelve la información más relevante de la Incidencia que lo invoque.

### **RutasActivity.java**

- *Activity\_rutas.xml*: Estructura de la lista de rutas.
- Método `getRutasUsuario(String idGuia)`: Hace la llamada a la base de datos y devuelve las rutas de los usuarios del guía que está manejando aplicación.

### **TemplateActivity.java**

- Método `onCreateOptionsMenu(Menu)`: Crea el menú común a todas las pantallas.
- Método `onOptionsItemSelected(MenuItem)`: Establece la acción al pulsar en una opción del menú.

### **Tramo.java**

- Método `Tramo(int,double,double,double,double,double, double, int)`: Constructor de un Tramo.
- Método `getIdTramo()`: Devuelve el identificador del Tramo que lo invoque.
- Método `setValoracion(double)`: Introduce el identificador en el Tramo que lo invoque.
- Método `getInicioLat()`: Devuelve la latitud del inicio del Tramo que lo invoque.
- Método `getInicioLng()`: Devuelve la longitud del inicio del Tramo que lo invoque.
- Método `getFinalLat()`: Devuelve la latitud del fin del Tramo que lo invoque.
- Método `getFinalLng()`: Devuelve la longitud del fin del Tramo que lo invoque.
- Método `getValoracionDer()`: Devuelve la valoración de la acera derecha del Tramo que lo invoque.
- Método `getValoracionIzq()`: Devuelve la valoración de la acera izquierda del Tramo que lo invoque.
- Método `getOrden()`: Devuelve el orden del Tramo que lo invoque.

### **Usuario.java**

- Método `Usuario(int, String, String, String, String, String, String, String, String, String, Guia)`: Constructor de un usuario.
- Método `getIdUsuario()`: Devuelve el identificador del Usuario que lo invoque.
- Método `setIdUsuario(int)`: Introduce el identificador en el Usuario que lo invoque.
- Método `getNombre()`: Devuelve la nombre del Usuario que lo invoque.
- Método `setNombre(String)`: Introduce el nombre en el Usuario que lo invoque.
- Método `getApellido1()`: Devuelve el primer apellido del Usuario que lo invoque.
- Método `setApellido1(String)`: Introduce el primer apellido en el Usuario que lo invoque.
- Método `getApellido2()`: Devuelve el segundo apellido Usuario que lo invoque.
- Método `setApellido2(String)`: Introduce el segundo apellido en el Usuario que lo invoque.

- Método `getDireccion()`: Devuelve la dirección del Usuario que lo invoque.
- Método `setDireccion(String)`: Introduce la dirección en el Usuario que lo invoque.
- Método `getLocalidad()`: Devuelve la localidad del Usuario que lo invoque.
- Método `setLocalidad(String)`: Introduce la localidad en el Usuario que lo invoque.
- Método `getProvincia()`: Devuelve la provincia del Usuario que lo invoque.
- Método `setProvincia(String)`: Introduce la provincia en el Usuario que lo invoque.
- Método `getTelefono1()`: Devuelve el teléfono principal del Usuario que lo invoque.
- Método `setTelefono1(String)`: Introduce el teléfono principal en el Usuario que lo invoque.
- Método `getTelefono2()`: Devuelve el teléfono secundario del Usuario que lo invoque.
- Método `setTelefono2(String)`: Introduce el teléfono secundario en el Usuario que lo invoque.
- Método `getEdad()`: Devuelve la edad del Usuario que lo invoque.
- Método `setEdad(String)`: Introduce la edad en el Usuario que lo invoque.
- Método `getGuia()`: Devuelve el Guia del Usuario que lo invoque.
- Método `setGuia(Guia)`: Introduce el Guia en el Usuario que lo invoque.
- Método `toString()`: Devuelve la información más relevante del Usuario que lo invoque.

### **UsuariosActivity.java**

- *Activity\_usuarios.xml*: Estructura que se compone de la lista de usuarios de un determinado guía.
- Método `infoUsuario()`: Devuelve toda la información del usuario que se ha seleccionado.
- Método `mensaje(String)`: Muestra por pantalla el mensaje indicado.
- Método `usuariosstatus(String)`: Hace una llamada a la base de datos y devuelve los usuarios del guía que esté manejando la aplicación.

### **Paquetes utilizados en el desarrollo de la aplicación**

- `Java.util`: Este paquete contiene el marco de las colecciones y clases de colecciones tradicionales.
- `Android.content`: Este paquete contiene clases para el acceso y la publicación de datos en el dispositivo.
- `Android.view`: Este paquete proporciona clases básicas de interfaz de usuario que se encargan del diseño de la pantalla y la interacción con el usuario.
- `Android.widget`: El paquete contiene elementos de la interfaz de usuario para usar en la pantalla de la aplicación.
- `Android.app`: Este paquete contiene clases de alto nivel que encapsulan el modelo general de la aplicación Android.

- `Android.os`: Este paquete proporciona servicios básicos del sistema operativo, el paso de mensajes y la comunicación entre procesos en el dispositivo.
- `Android.util`: Proporciona métodos de utilidad común, métodos de conversión y utilidades XML.
- `Android.graphics`: Este paquete proporciona herramientas de gráficos de bajo nivel, como lienzos, filtros de color, puntos y rectángulos que permiten dibujar en la pantalla directamente.
- `Android.location`: Este paquete contiene las clases del framework API que definen los servicios basados en la localización de Android.
- `Android.net`: Este paquete proporciona las clases para el acceso a la red.
- `Org.apache.http`: Este paquete se ocupa de las cosas fundamentales que requiere utilizar el protocolo HTTP.
- `Com.google.android.gms.maps`: Este paquete contiene los mapas de Google de la API de Android.



## Capítulo 5: Conclusiones y trabajos futuros

Cabe destacar que la colaboración que hemos tenido en este proyecto con la *Fundación ONCE* nos ha ayudado a aprender mucho acerca de cómo se debe y se deberían desarrollar todas las aplicaciones móviles, para que sean totalmente accesibles, y la ínfima cantidad de ellas que realmente lo son.

Es gratificante ver cómo, con unas nociones muy básicas de accesibilidad, hemos podido desarrollar dos aplicaciones móviles que responden correctamente al activar el modo accesible para discapacitados visuales.

Aunque una gran barrera para un usuario o tutor con deficiencia visual son los mapas. No encontramos ninguna empresa que ofreciera un mapa accesible en su totalidad, centrándonos en este tipo de usuarios. Y por ello sería muy útil desarrollar un sistema de lectura de mapas y manejo de estos por medio de comandos vocales, con el que se pudiera configurar una ruta o navegar por un mapa sin tener contacto visual con el dispositivo.

Respecto a otra posible ampliación de este proyecto, está la de extender las arañas de movilidad. Creemos que el éxito de estas herramientas se basa en que sobre todo sea útil en centros de discapacitados, por ello entendemos que dedicar una temporada a la promoción y tutorización para su correcto uso y a recibir de ello *feedback* sería muy beneficioso para la mejora y propagación del proyecto.





# Bibliografía

- [1] Autor: *Developers Google* [en línea]. [Consulta: Febrero 2014].  
<https://developers.google.com/maps/documentation/javascript/?hl=es>
- [2] Autor: Developers Google [en línea]. [Consulta: Mayo 2014].  
<https://developers.google.com/maps/documentation/android/?hl=es>
- [3] Autor: anidocs [en línea]. [Consulta: Abril 2014].  
<http://www.anidocs.es/bootstrap/docs/index.php>
- [4] Autor: W3C [en línea]. [Consulta: Abril 2014]. <http://www.w3.org/>
- [5] Autor: myphp [en línea]. [Consulta: Abril 2014]. <http://mx1.php.net/>
- [6] Autor: open source project [en línea]. [Consulta: Mayo 2014].  
<http://developer.android.com/>
- [7] Autor: FAMMA [en línea]. [Consulta: Noviembre 2013].  
<http://www.famma.org/accessibility>
- [8] Autor: Google [en línea]. [Consulta: Noviembre 2013].  
<https://play.google.com/store/apps/details?id=com.SafetyGPS&hl=es>
- [9] Autor: SafetyGPS [en línea]. [Consulta: Noviembre 2013]. <http://www.safetygps.com>
- [10] Autor: Google [en línea]. [Consulta: Noviembre 2013].  
<https://play.google.com/store/apps/details?id=com.sloper.accesibilidad&hl=es>



## Anexo 1: Definiciones, acrónimos y abreviaturas

- **Maker:** Es un marcador que indica ubicaciones individuales en el mapa.
- **Usuario:** Persona discapacitada.
- **Incidencia:** Indicación de un obstáculo o peligro en el camino que el usuario marca para que el guía encargado cambie y trate de evitar en la ruta. Al ser algo que el usuario marca de forma rápida (por eso no se da opción a comentario), si al tutor no le quedara claro el porqué de esta incidencia, debería contactar con el usuario que la reportó.
- **Inseguridad:** Este tipo de incidencia abarca toda situación en la cual el usuario pueda sentirse incómodo, tales como una pelea, aglomeración, etc.
- **Guía/Tutor:** Persona que conduce, dirige, aconseja u orienta al usuario.
- **Obstáculo:** Este tipo de incidencia comprende cualquier tipo de impedimento que el usuario pueda tener que le moleste en su recorrido.
- **API:** Interfaz de programación de aplicaciones (IPA) o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.
- **SDK:** kit de desarrollo de software (siglas en inglés de software development kit) es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador crear aplicaciones para un sistema concreto, por ejemplo ciertos paquetes de software, frameworks, plataformas de hardware, computadoras, videoconsolas, sistemas operativos, etc.



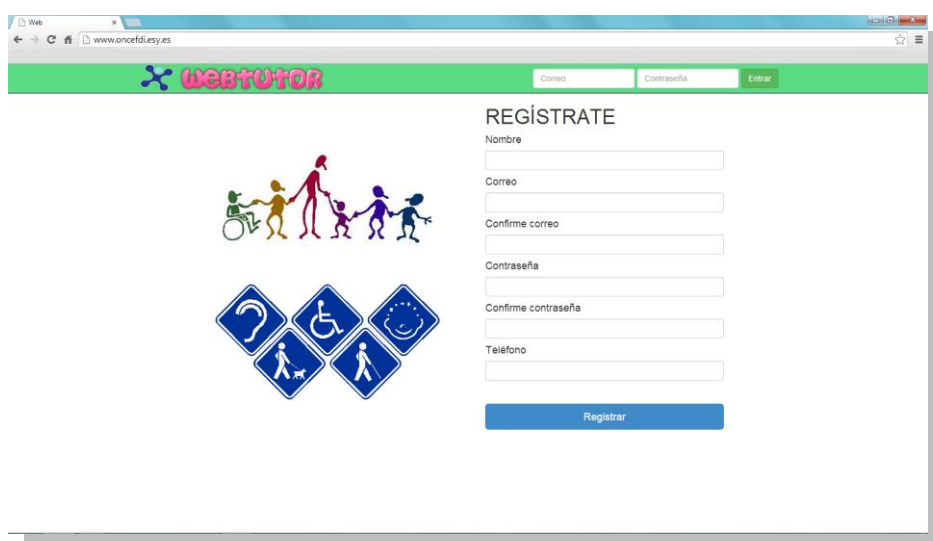
## Anexo 2: Manual de usuario

En este manual se detallará la instalación de las aplicaciones (MiTutor y SoyTutor) y el uso de las mismas y de la página Web (Webtutor), acompañado de imágenes para facilitar la comprensión.

### Página web (WebTutor)

#### Acceso

Abrir el navegador instalado previamente, introducimos la dirección de la página Web [www.oncedi.esy.es](http://www.oncedi.esy.es)



*Ilustración 52: Manual Página web: Acceso.*

#### Registro

Para acceder a las funcionalidades de la página es necesario estar registrado en el sistema. Para registrar un nuevo guía hay que rellenar todos los campos del formulario y pulsar el botón “Registrar”.

Los campos del formulario son:

- Nombre: el nombre del guía.
- Correo: correo con el cual posteriormente podrá iniciar la sesión.
- Confirmar Correo: se trata de introducir el mismo correo que en el campo anterior para validar y evitar posibles errores.
- Contraseña: contraseña que servirá para iniciar sesión.
- Confirmar contraseña: debe ser la misma contraseña que en el campo anterior para evitar errores.
- Teléfono: debe ser el número de móvil en el que va a estar instalada la aplicación SoyTutor.

*Ilustración 53: Manual Página web: Registro.*

## Inicio sesión

Una vez realizado el registro, podrá iniciar sesión en la página introduciendo el correo y la contraseña proporcionado en el registro y pulsando el botón “Entrar”.

*Ilustración 54: Manual Página web: Inicio sesión.*

## Menú principal

Desde el menú principal podemos acceder a las principales funcionalidades de la página Web. Se compone de tres bloques principales: usuarios (encontrará todo lo relacionado con los usuarios del guía), rutas (encontrará todo lo relacionado con las rutas de los usuarios del guía) y configuración.



Ilustración 55: Manual Página web: Menú usuarios.

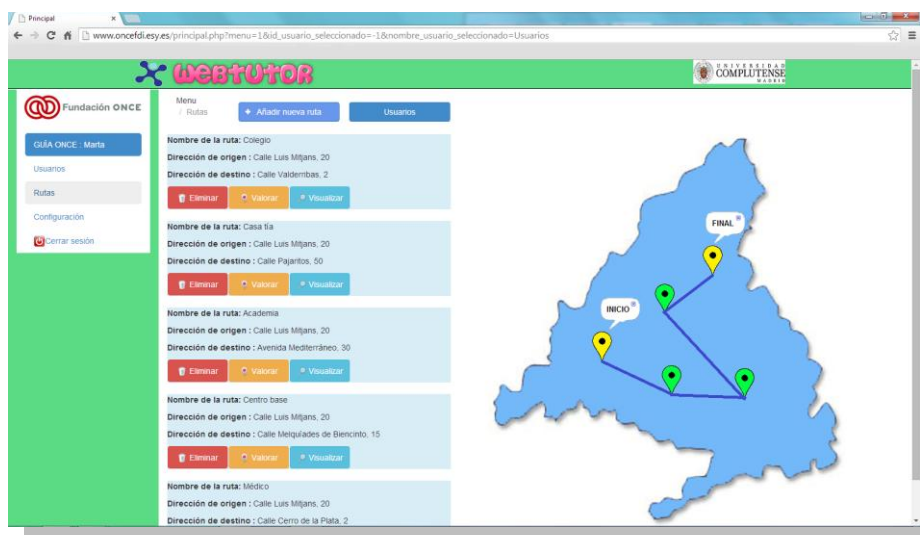


Ilustración 56: Manual Página web: Menú Rutas.

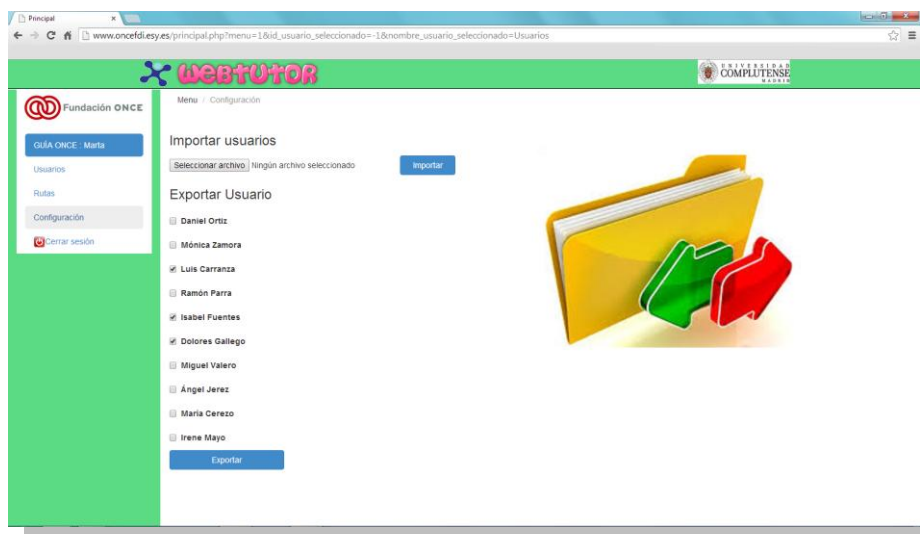


Ilustración 57: Manual Página web: Menú configuración.

## Añadir nuevo usuario

El botón “Añadir nuevo usuario” permite incorporar nuevos usuarios. Para ello, debe rellenar todos los campos del formulario y pulsar el botón “Registrar Usuario”.

Datos del formulario:

- Nombre: debe introducir el nombre del nuevo usuario para posteriormente identificar las rutas asociadas al mismo.
- Primer apellido: debe introducir el primer apellido del nuevo usuario.
- Segundo apellido: debe introducir el segundo apellido del nuevo usuario.
- Dirección: debe introducir la dirección del domicilio del nuevo usuario.
- Localidad: debe introducir la localidad de la dirección del domicilio del nuevo usuario.
- Provincia: debe introducir la provincia correspondiente a la dirección y localidad introducida anteriormente.
- Edad: debe introducir la edad del nuevo usuario.
- Teléfono del usuario: debe introducir el número de móvil en el que va instalada la aplicación MiTutor, para que el guía pueda realizar llamadas si le ocurre algún incidente al usuario.
- Teléfono de contacto prioritario: debe introducir el número de teléfono con el que el usuario se pueda poner en contacto en caso de tener un incidente.
- Teléfono de contacto secundario: debe introducir el número de teléfono con el que el usuario se pueda poner en contacto en caso de incidente y el teléfono prioritario no se encuentre disponible.

The screenshot shows a web browser window with the URL [www.oncefdi.es/usuario.php?id=20](http://www.oncefdi.es/usuario.php?id=20). The page has a green header with the 'WEBTUTOR' logo and the 'UNIVERSIDAD COMPLUTENSE MADRID' logo. The main content area is a light green box titled 'Registro nuevo usuario'. It contains a form with the following fields and values:

Field	Value
Nombre	Daniel
Edad	16
Primer apellido	Ortiz
Segundo apellido	Sainz
Dirección (Calle y número)	Calle Luis Miñans
Localidad	Madrid
Provincia	Madrid
Teléfono del usuario	675631654
Teléfono de contacto prioritario	621483200
Teléfono de contacto secundario	671328466

A blue button labeled 'Registrar Usuario' is located at the bottom right of the form.

*Ilustración 58: Manual Página web: Añadir nuevo usuario.*

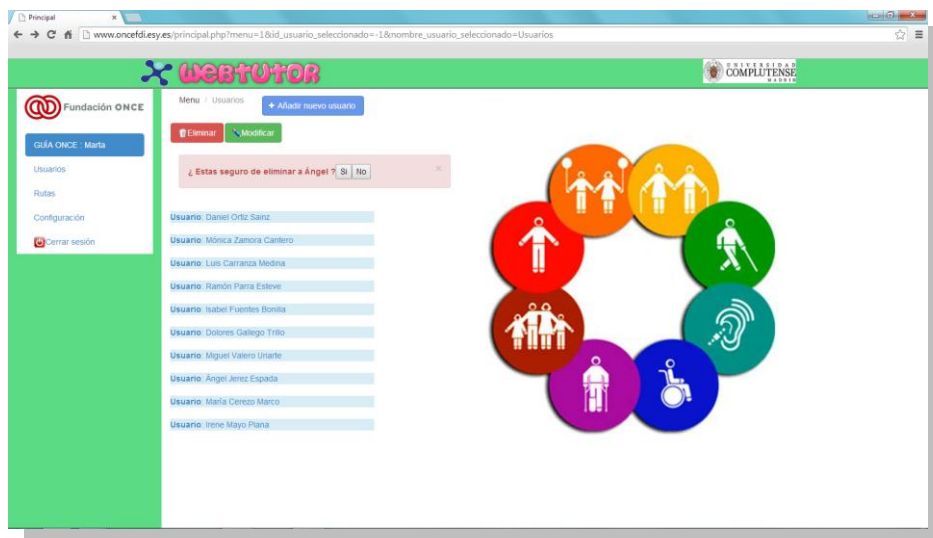


## Eliminar Usuario

El botón “Eliminar” dentro del menú usuarios permite eliminar un usuario. Para ello se desplegará una lista con los usuarios del guía. Al seleccionar el usuario se pedirá confirmación para la eliminación del mismo.



*Ilustración 59: Manual Página web: Elegir usuario a eliminar.*



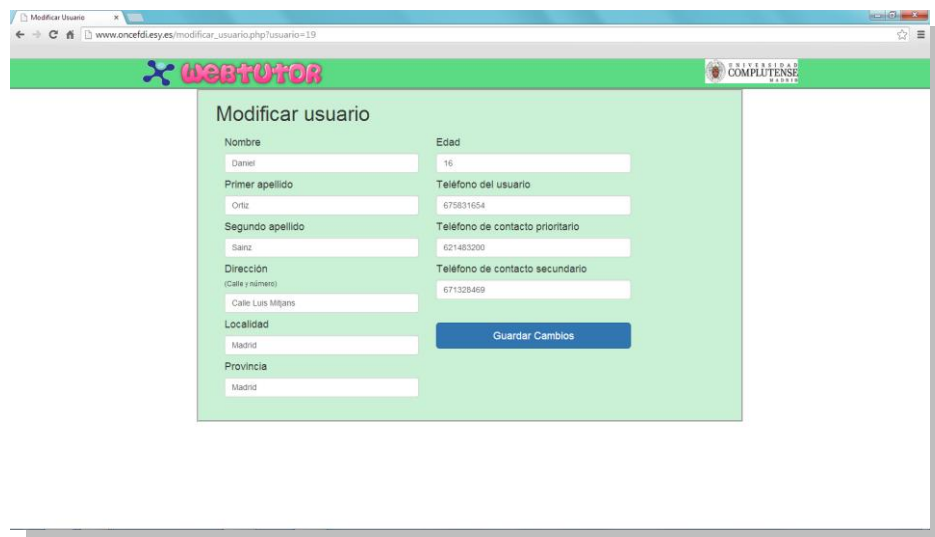
*Ilustración 60: Manual Página web: Confirmar eliminar usuario seleccionado.*

## Modificar usuario

El botón “Modificar” del menú de usuarios desplegará una lista con los usuarios del guía. Al seleccionar el que se desea modificar, se mostrará un formulario con los datos introducidos en el registro, permitiendo modificar los datos. Pulsando el botón “Guardar Cambios” el usuario quedará modificado.



*Ilustración 61: Manual Página web: Seleccionar usuario a modificar.*



*Ilustración 62: Manual Página web: Modificar usuario seleccionado.*

## Visualizar usuario

Haciendo Click sobre un usuario del menú de usuarios podrá ver toda la información introducida acerca del mismo.

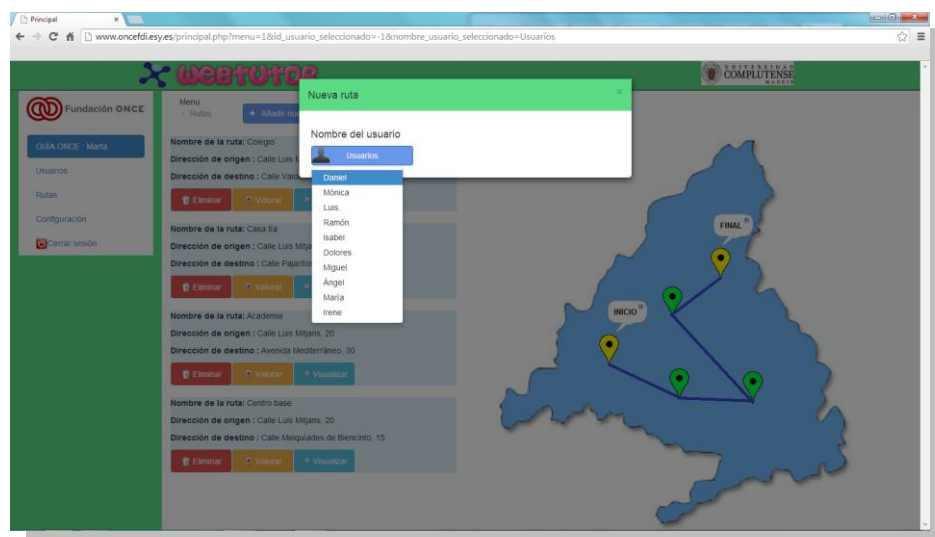


*Ilustración 63: Manual Página web: Visualizar información de un usuario.*

## Añadir nueva ruta

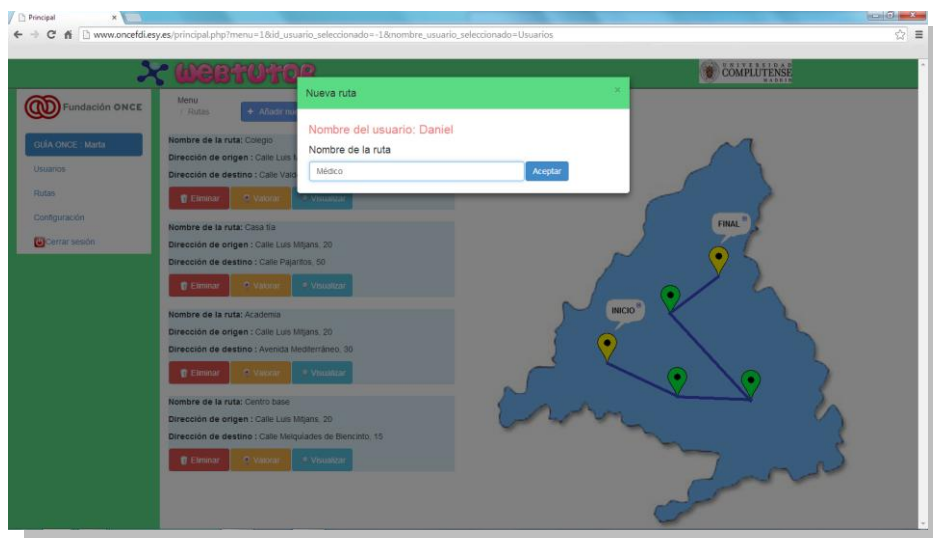
El botón “Añadir nueva ruta” permite incorporar nuevas rutas asociadas a usuarios ya añadidos en el sistema.

En primer lugar debe introducir el usuario para el que se va a configurar la ruta mediante un desplegable.



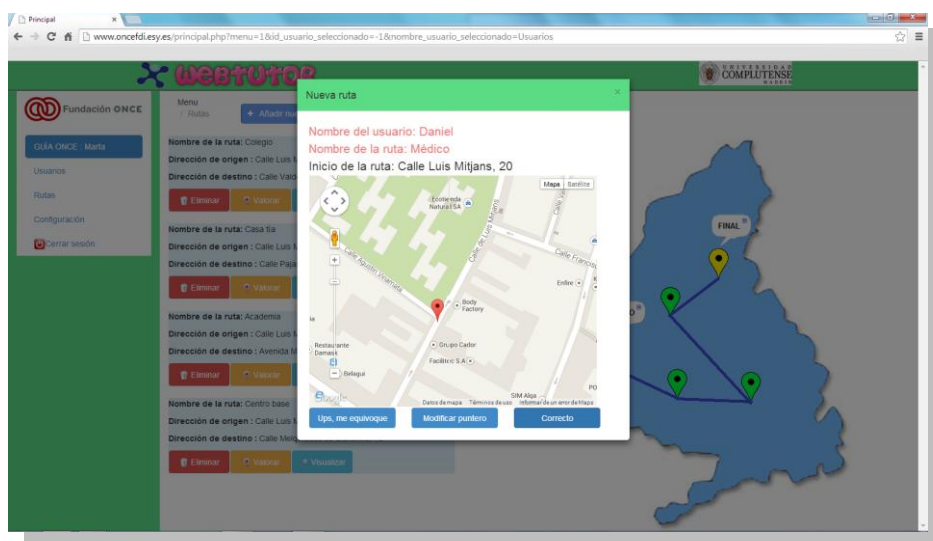
*Ilustración 64: Manual Página web: Seleccionar usuario para configurar la nueva ruta.*

En segundo lugar pondrá el nombre que se le va a dar a la nueva ruta.



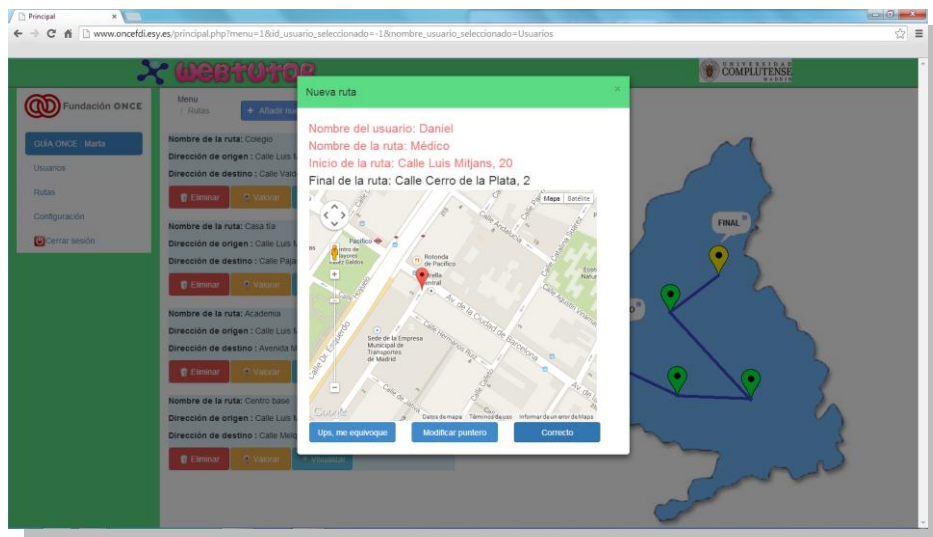
*Ilustración 65: Manual Página web: Insertar nombre de la nueva ruta.*

En tercer lugar introduce la dirección de inicio de la ruta. Se mostrará esta dirección en un mapa que debe configurar (modificando dirección completa o moviendo el marker) para que sea la deseada y una vez finalizada pulsar botón “Correcto”.



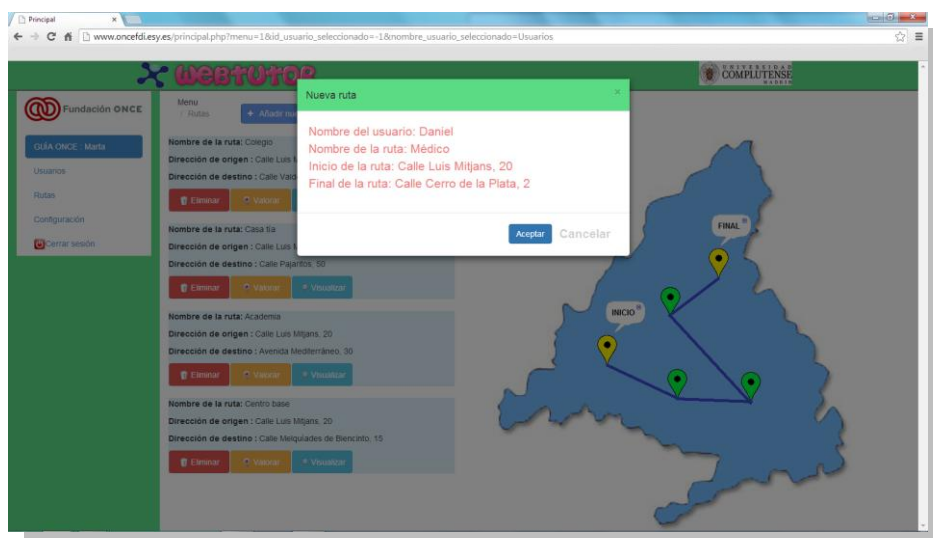
*Ilustración 66: Manual Página web: Insertar inicio de la ruta.*

En cuarto lugar introduce la dirección de final de la ruta, se realiza el mismo proceso que en la dirección de inicio.



*Ilustración 67: Manual Página web: Insertar final de la ruta.*

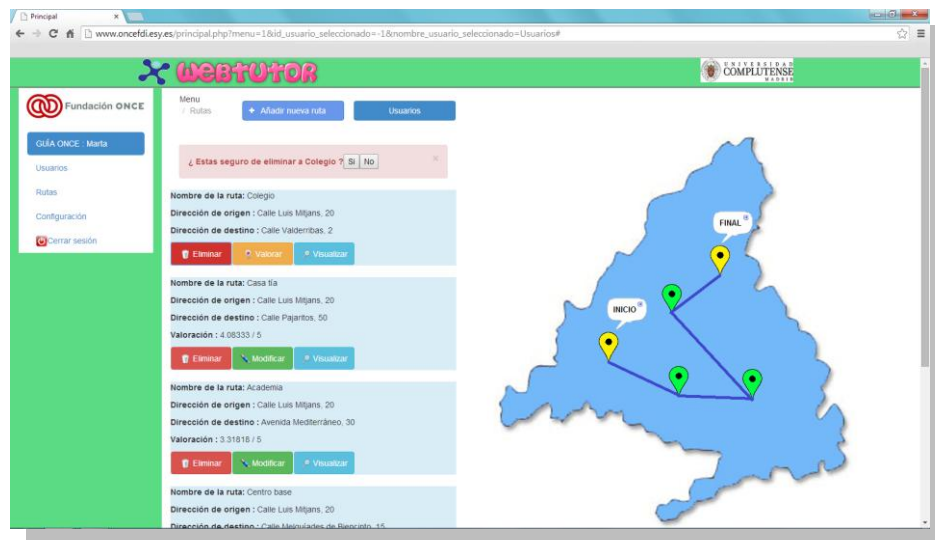
Por último solo debemos Aceptar y la ruta quedará añadida y asignada al usuario en la base de datos.



*Ilustración 68: Manual Página web: Añadir nueva ruta.*

## Eliminar ruta

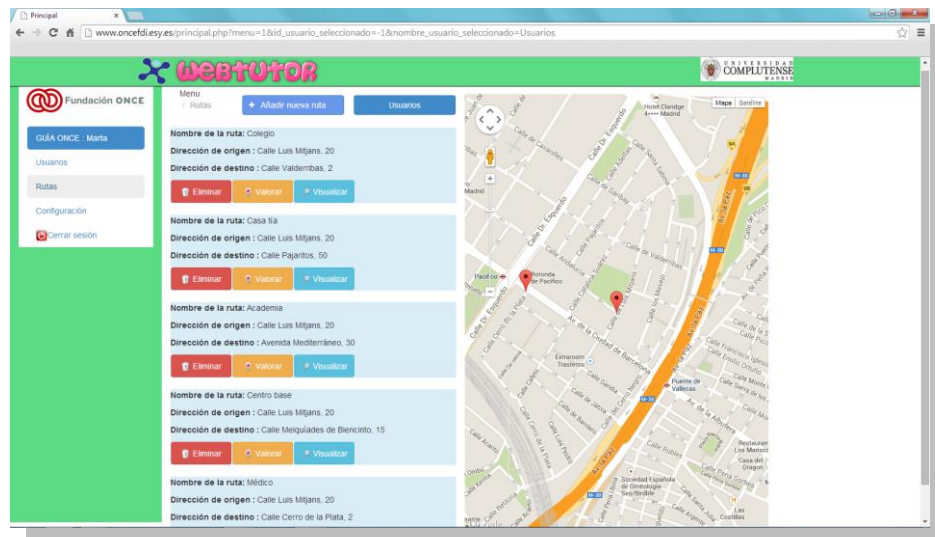
El botón “Eliminar” del menú de rutas, eliminará la ruta de la base de datos tras una confirmación.



*Ilustración 69: Manual Página web: Eliminar ruta.*

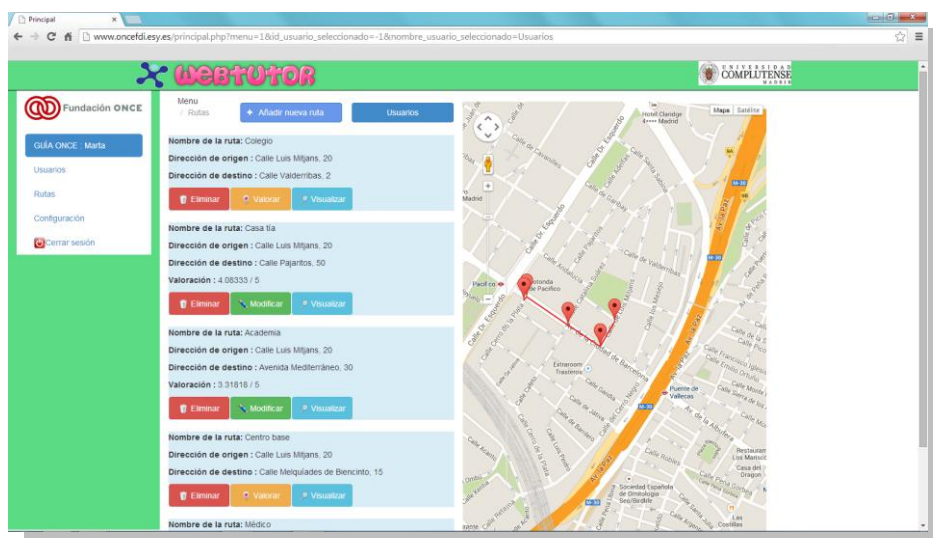
## Visualizar una ruta

Pulsando el botón “Visualizar” podemos ver el inicio y el final de la ruta y los tramos de la misma, en el caso de estar valorada, en un mapa.



*Ilustración 70: Manual Página web: Visualizar ruta sin valorar.*

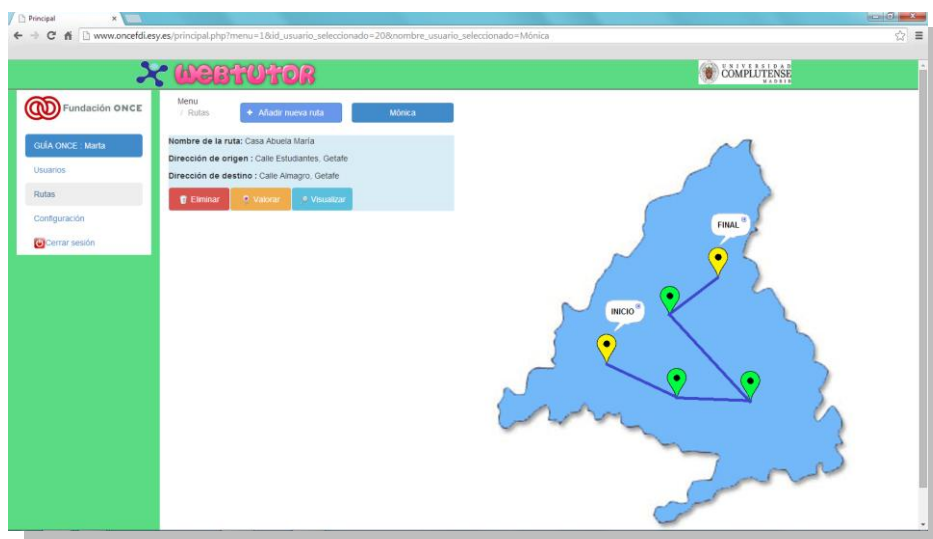




*Ilustración 71: Manual Página web: Visualizar ruta valorada.*

## Visualizar rutas de un usuario

Pulsando el botón “Usuarios”, se desplegará una lista de los usuarios del guía y seleccionando uno de ellos podemos visualizar todas las rutas asignadas a este usuario.

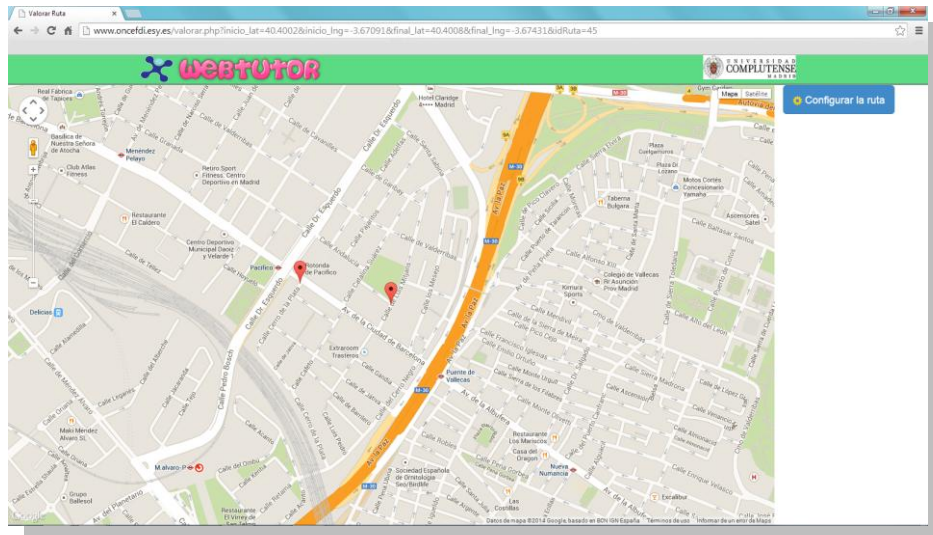


*Ilustración 72: Manual Página web: Visualizar rutas de un usuario.*

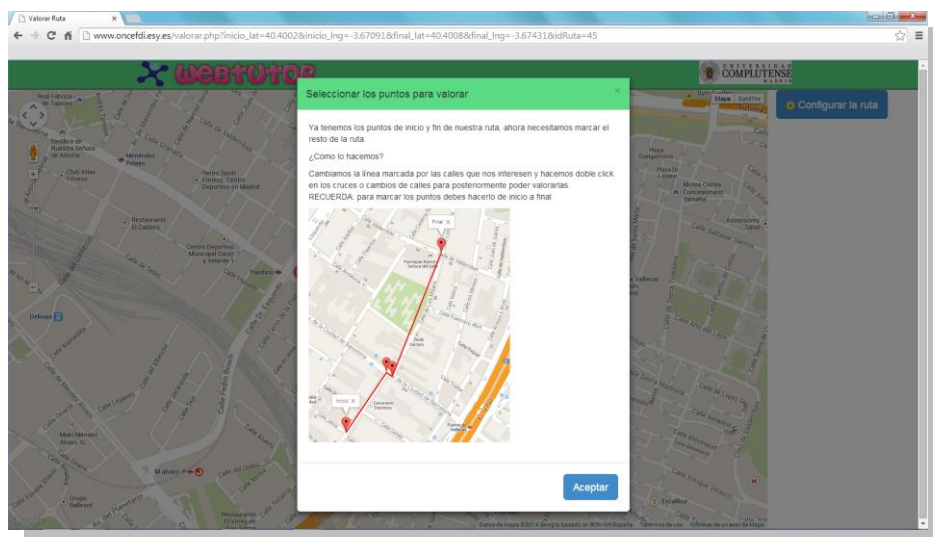
## Valorar ruta

Pulsando el botón “Valorar”, se mostrará un mapa con los puntos de inicio y final de la ruta.

Pulsando el botón “Configurar la ruta”, se mostrará una breve explicación de lo que deberá hacer para valorar la ruta, una vez leído como se realiza la valoración, pulsamos el botón “Aceptar” y se procede a realizar la configuración de los tramos de la ruta.



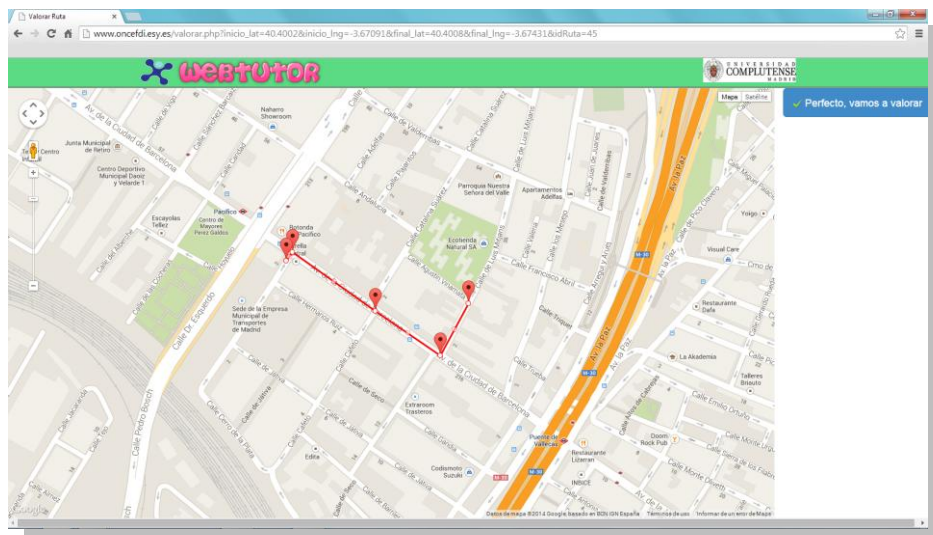
*Ilustración 73: Manual Página web: Valorar una ruta.*



*Ilustración 74: Manual Página web: Instrucciones de cómo construir una ruta.*

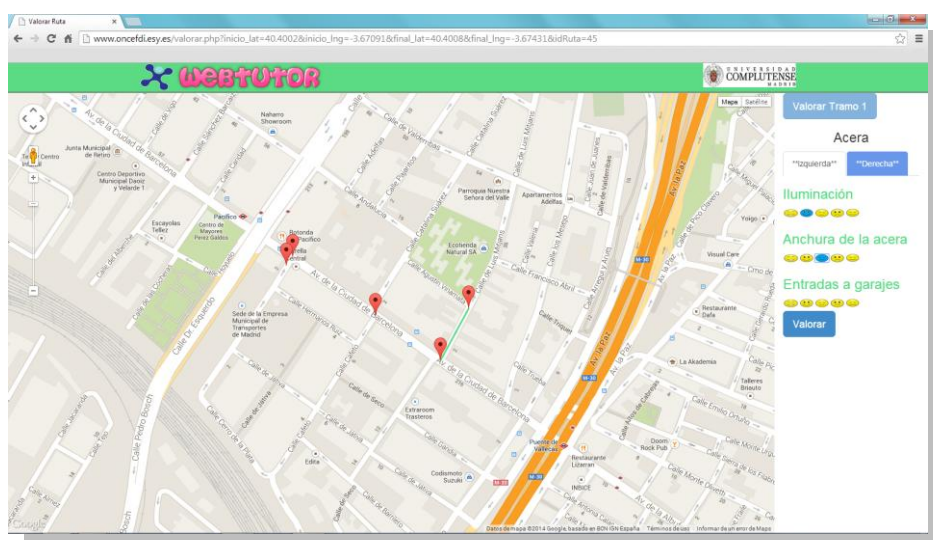
Un vez finalizada la configuración de la ruta pulsar el botón “Perfecto, vamos a valorar”, donde se procederá a valorar los tramos por aceras mediante los factores (Iluminación, Anchura de la acera y Entradas a garajes).





*Ilustración 75: Manual Página web: Configurar una ruta a valorar.*

Una vez valorados todos los tramos insertados pulsar botón “guardar” y la ruta queda valorada en la base de datos.

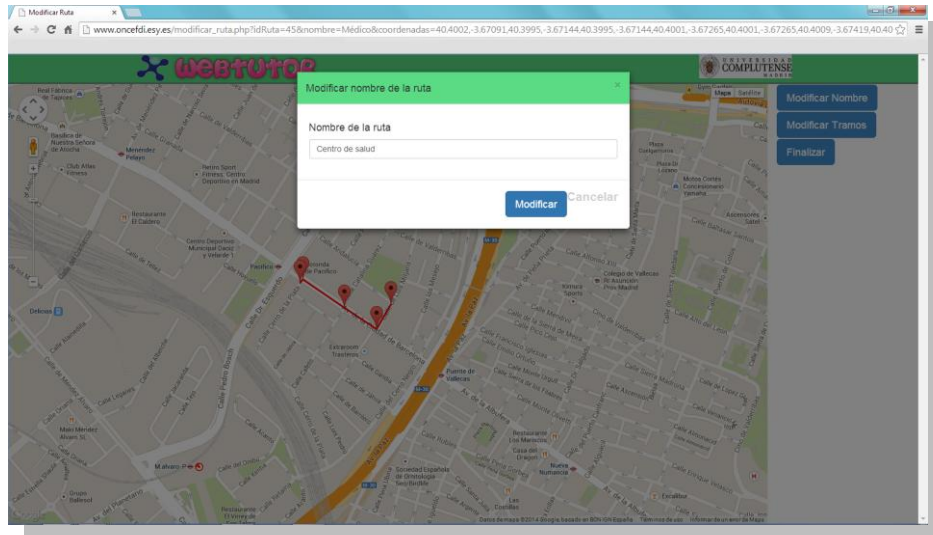


*Ilustración 76: Manual Página web: Valorar tramos de la ruta por aceras.*

## Modificar ruta

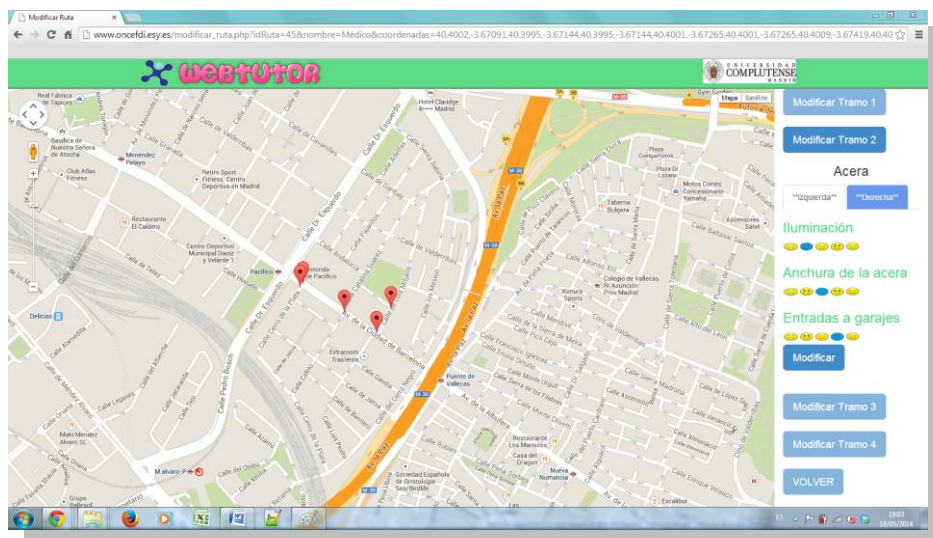
Pulsando el botón “Modificar” podrá:

Modificar el nombre de la ruta y pulsar el botón “Modificar”.



*Ilustración 77: Manual Página web: Modificar nombre de la ruta.*

Modificar las valoraciones de los tramos deseados. Para ello se pulsará el tramo deseado, volviendo a valorar las aceras. Una vez finalizado el proceso, pulsando el botón “Modificar” y las modificaciones quedarán guardadas en la base de datos.



*Ilustración 78: Manual Página web: Modificar valoración de los tramos de la ruta.*

Pulsando el botón “Finalizar” volvemos al menú de rutas.

## Exportar usuarios

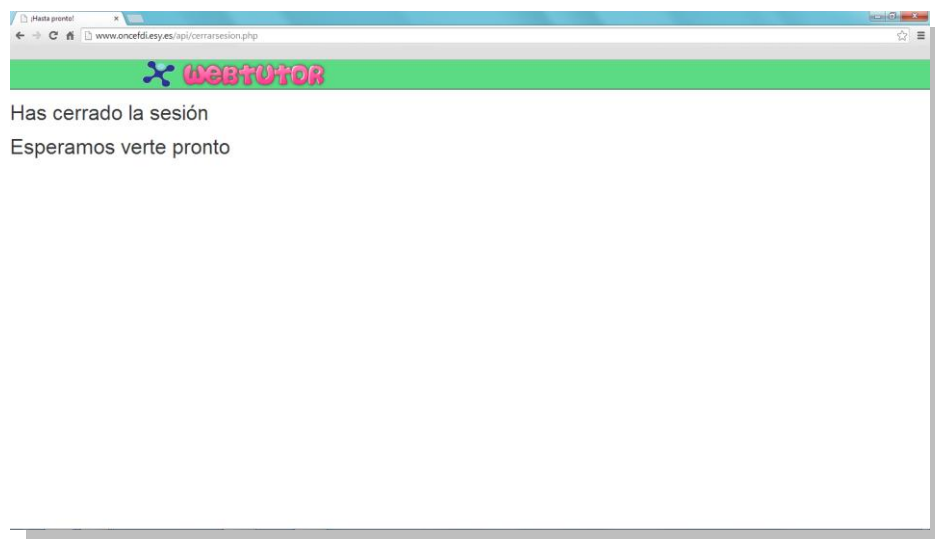
En el menú de configuración podrá seleccionar los usuarios que se desean exportar y pulsar el botón “Exportar”. El proceso genera un archivo que debemos entregar a otro guía para que éste pueda añadir los usuarios como suyos.

## Importar usuarios

En el menú configuración pulsar el botón “Seleccionar archivo” donde introducirá el archivo recibido de otro guía. Una vez introducido el archivo pulsar el botón “Importar” y los usuarios contenidos en el archivo serán añadidos a sus usuarios.

## Cerrar sesión

Al pulsar el botón “Cerrar sesión” el guía cierra la sesión para poder iniciar otra sesión con otro guía.



*Ilustración 79: Manual Página web: Cerrar sesión.*

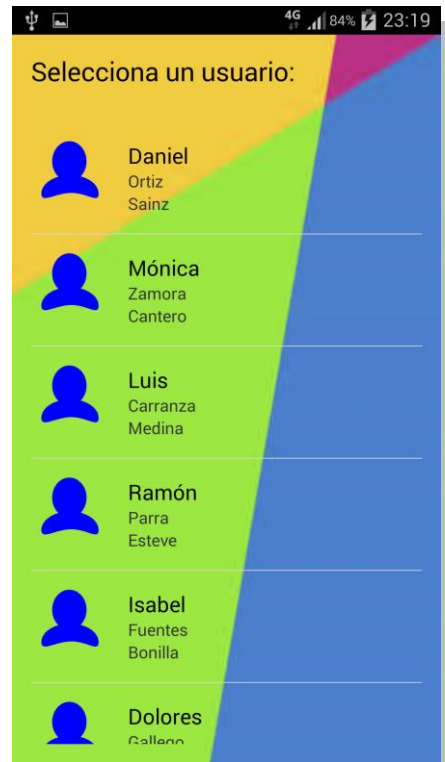
## Aplicación del usuario (MiTutor)

### Inicio sesión

Una vez realizado el registro a través de la web, se podrá realizar el acceso a la aplicación móvil MiTutor utilizando los datos de registro del guía (correo y contraseña). Una vez introducidos los datos del guía, la aplicación le mostrara todos los usuarios que dispone y tendrá que seleccionar el usuario para el que configurara la aplicación. Una vez seleccionado el usuario, la aplicación guardará esta información para no tenerla que solicitar cada vez que se inicie la aplicación.



*Ilustración 80: Manual Aplicación usuario:  
Iniciar sesión.*



*Ilustración 81: Manual Aplicación usuario:  
Selección usuario.*

## Menú Principal

Desde el menú principal podremos acceder a las principales funcionalidades de la aplicación: “iniciar ruta” (podrá iniciar una de las rutas que tenga guardadas), “llamada tutor” (realizara una llamada telefónica a su tutor), “incidencias” (podrá reportar una incidencia que haya tenido), “configuración” (podrá realizar algunos cambios en los ajustes de la aplicación), “información” (podrá ver información sobre la aplicación), y “salir” (podrá salir de la aplicación).



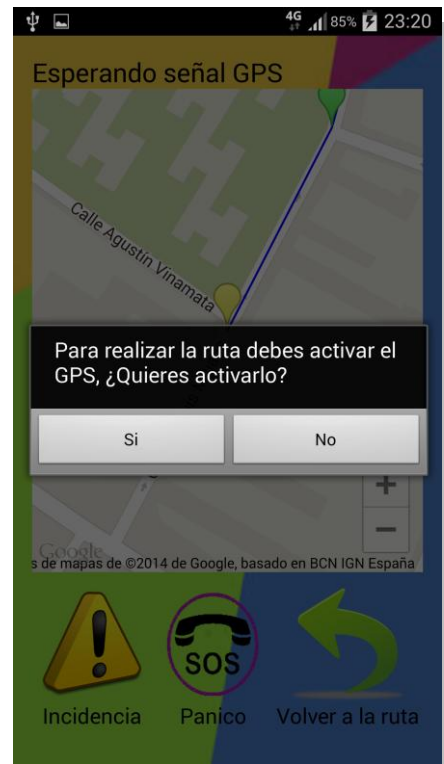
*Ilustración 82: Manual Aplicación usuario: Menú principal.*

## Iniciar Ruta

Cuando el usuario inicie una ruta, la aplicación le mostrara las rutas que tiene guardadas y el usuario tendrá que seleccionar la ruta que desea iniciar, una vez seleccionada la aplicación cargara la ruta y en caso de no tener el GPS activado le solicitara mediante un mensaje por pantalla si desea activarlo para que le pueda guiar.



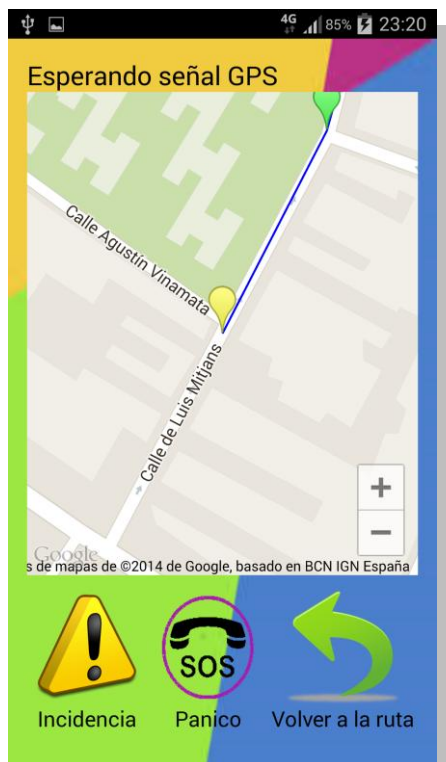
*Ilustración 83: Manual Aplicación usuario: Selección ruta.*



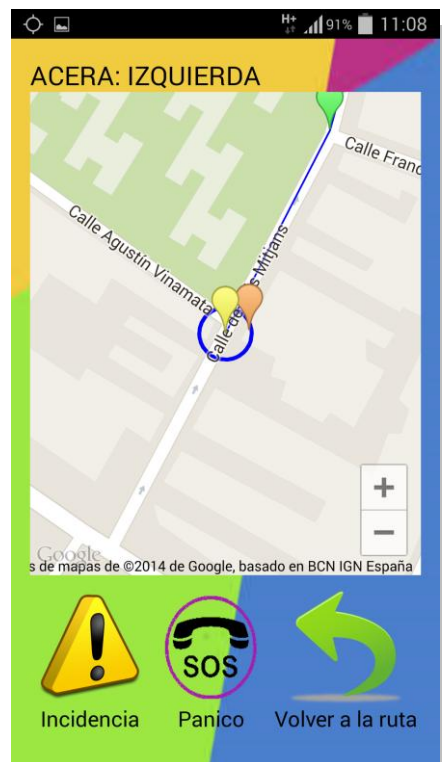
*Ilustración 84: Manual Aplicación usuario: Activar GPS.*

Una vez activado el GPS el usuario deberá esperar a que tenga señal GPS. Una vez tenga señal GPS la aplicación comenzara a guiarle, indicándole porque acera debe ir en cada momento (según lo configurado al crear la ruta en la web). En cualquier momento el usuario tendrá a su disposición la posibilidad de marcar una incidencia, realizar una llamada de emergencia o volver a la ruta.





*Ilustración 85: Manual Aplicación usuario:  
Esperando GPS.*



*Ilustración 86: Manual Aplicación usuario:  
Guiando al usuario.*

## Volver a la ruta

En caso de que el usuario se perdiera o saliese de la ruta, podría volver a ella utilizando la opción volver a la ruta.

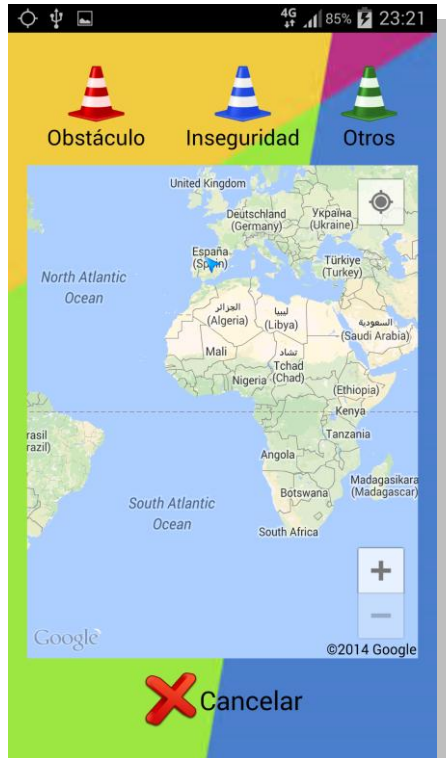


*Ilustración 87: Manual Aplicación usuario: Volver a la ruta.*

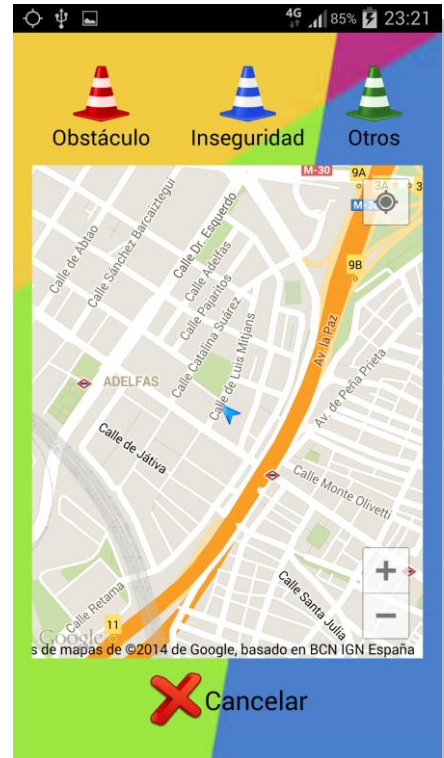


## Incidencia

Cuando el usuario decide reportar una incidencia, la aplicación le mostrara un mapa, en el cual podrá acercarse a la posición actual mediante el botón de la esquina superior del mapa. Para reportar la incidencia el usuario deberá tener señal GPS para que las coordenadas de la incidencia sean lo más exactas posibles y deberá seleccionar el tipo de incidencia a reportar (obstáculo, inseguridad, otros).



*Ilustración 88: Manual Aplicación usuario:  
Mapa incidencia.*



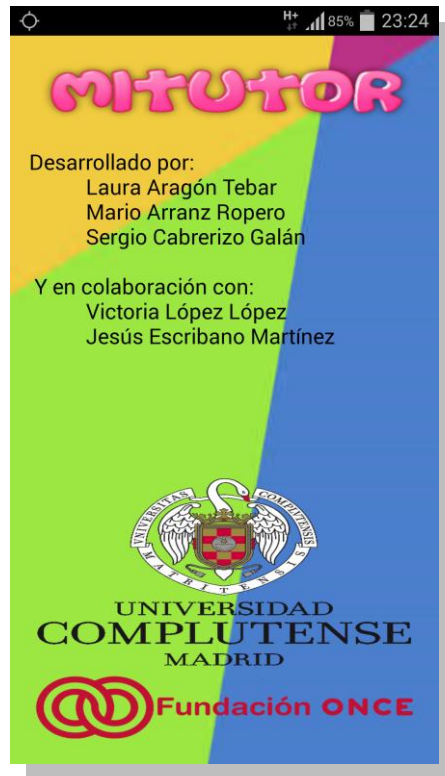
*Ilustración 89: Manual Aplicación usuario:  
Mapa incidencia 2.*

## Llamada Pánico

Durante la ruta si el usuario tuviera una emergencia dispone de un botón de Pánico (SOS) el cual realizaría una llamada telefónica al número de emergencia que tenga configurado.

## Información de la aplicación

El usuario podrá ver quienes desarrollaron la aplicación y con quien colaboraron.



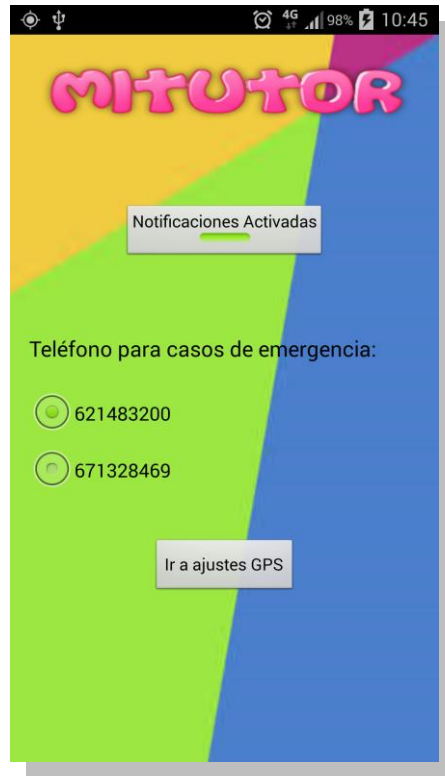
*Ilustración 90: Manual Aplicación usuario: Información.*

## Llamada tutor

Desde el menú principal el usuario tenía la opción de llamar a su tutor por si quisiera comunicarse con él, en la cual se realizara una llamada.

## Configuración

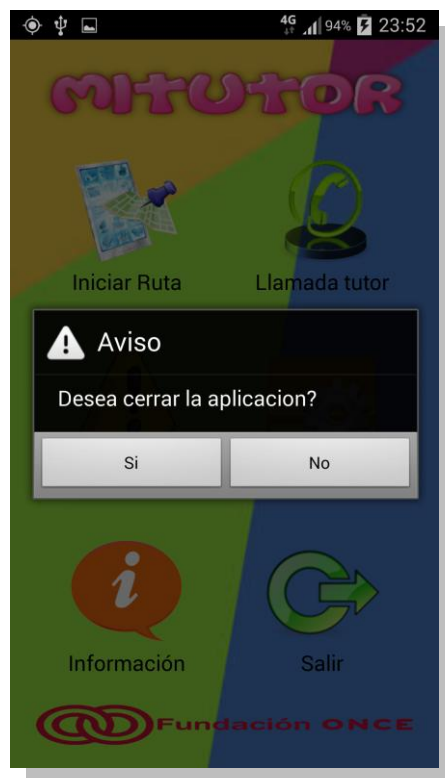
El usuario podrá modificar el teléfono que utilizara para los casos de emergencia, podrá ir a la configuración del GPS del móvil para activarla sino la tuviese y podrá desactivar o activar el envío de notificaciones.



*Ilustración 91: Manual Aplicación usuario: Configuración.*

## Cerrar Sesión

Utilizando esta opción el usuario cerrara la aplicación.



*Ilustración 92: Manual Aplicación usuario: Cerrar aplicación.*

## Aplicación del guía (SoyTutor)

### Iniciar Sesión

Una vez realizado el registro a través de la web, se podrá realizar el acceso a la aplicación móvil SoyTutor utilizando los datos de registro del guía (correo y contraseña). Una vez introducidos los datos del guía, la aplicación guardará esta información para no tenerla que solicitar cada vez que se inicie la aplicación.



*Ilustración 93: Manual Aplicación guía: Inicio sesión.*

## Menú Principal

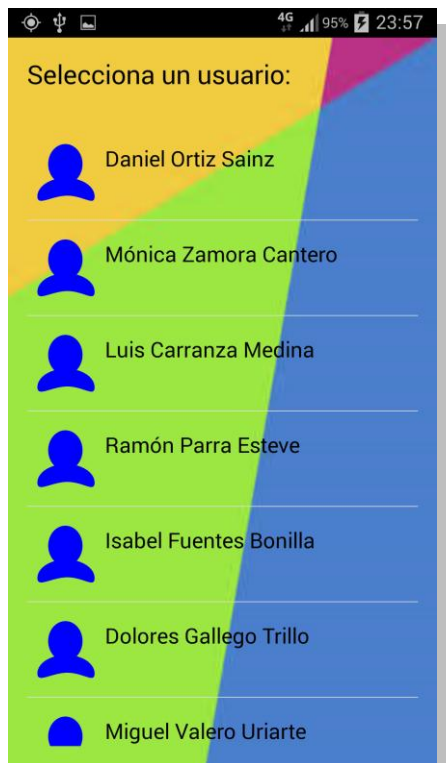
Desde el menú principal podremos acceder a las principales funcionalidades de la aplicación: “usuarios” (podrá ver la información de los usuarios que tiene a su cargo), “rutas usuarios” (podrá ver las rutas asociadas a sus usuarios), “incidencias” (podrá ver las incidencias marcadas por sus usuarios), “configuración” (podrá realizar algunos cambios en los ajustes de la aplicación), “información” (podrá ver información sobre la aplicación), y “salir” (podrá salir de la aplicación).



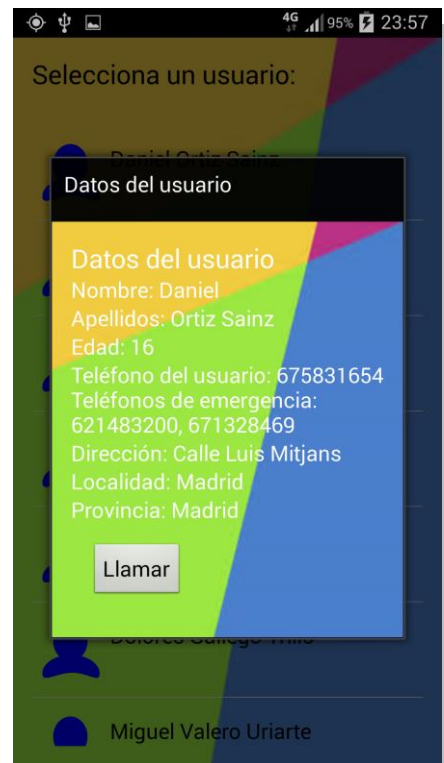
*Ilustración 94: Manual Aplicación guía: Menú principal.*

## Usuarios

Cuando selecciona usuarios en el menú principal, la aplicación le mostrara una lista con todos los usuarios que tiene el guía a su cargo, si se selecciona un usuario de la lista, se mostrara la información a dicho usuario y tendrá la opción de llamarlo.



*Ilustración 95: Manual Aplicación guía:  
Selección usuario.*



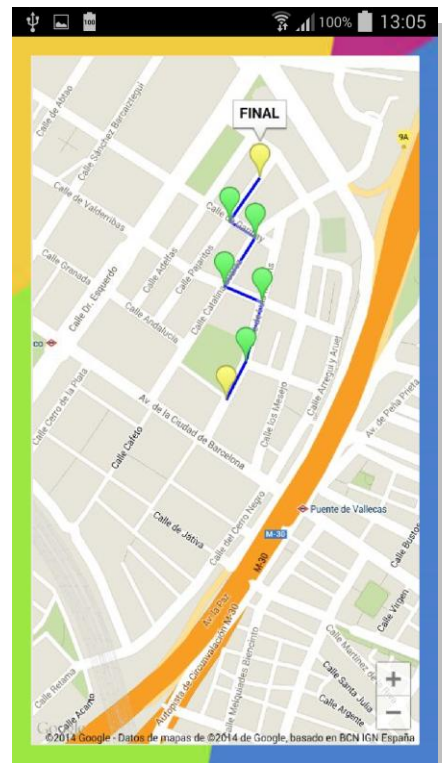
*Ilustración 96: Manual Aplicación guía:  
Datos del usuario.*

## Rutas Usuarios

El guía podrá visualizar las rutas de sus usuarios utilizando la opción “rutas usuarios” del menú principal, en este caso la aplicación le mostrará la lista de rutas asociadas a sus usuarios y si selecciona una podrá visualizarla.



*Ilustración 97: Manual Aplicación guía: Selección ruta.*



*Ilustración 98: Manual Aplicación guía: Visualización ruta.*



## Incidencias

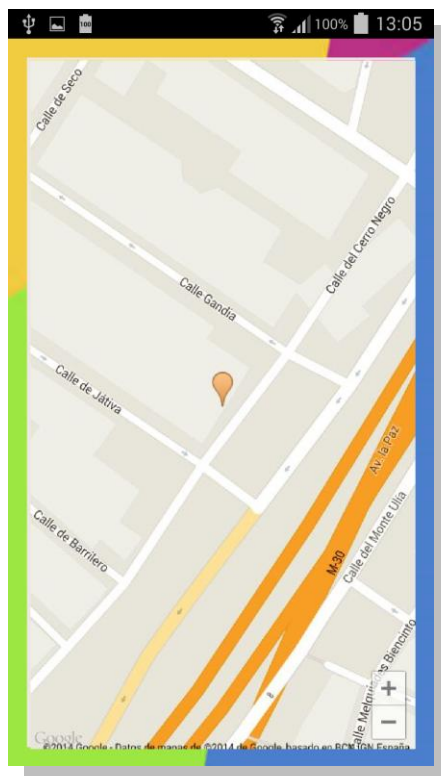
El guía podrá visualizar las incidencias de sus usuarios utilizando la opción “incidencias” del menú principal, en este caso la aplicación le mostrará la lista de incidencias asociadas a sus usuarios y si selecciona una podrá visualizarla. En la lista aparecerán los distintos tipos de incidencia y además se marcará la diferencia entre las vistas y no vistas por el usuario.



*Ilustración 99: Manual Aplicación guía: Lista incidencias.*



*Ilustración 100: Manual Aplicación guía: Lista incidencias 2.*



*Ilustración 101: Manual Aplicación guía: Visualización incidencia.*

También es posible acceder a esta funcionalidad cuando el guía recibe una notificación de incidencia.



*Ilustración 102: Manual Aplicación guía: Notificación incidencia.*

## Información de la Aplicación

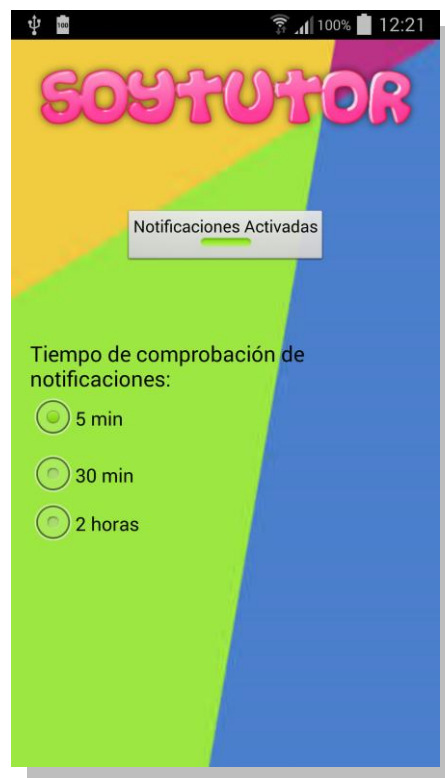
El guía podrá ver quienes desarrollaron la aplicación y con quien colaboraron.



*Ilustración 103: Manual Aplicación guía: Información.*

## Configuración

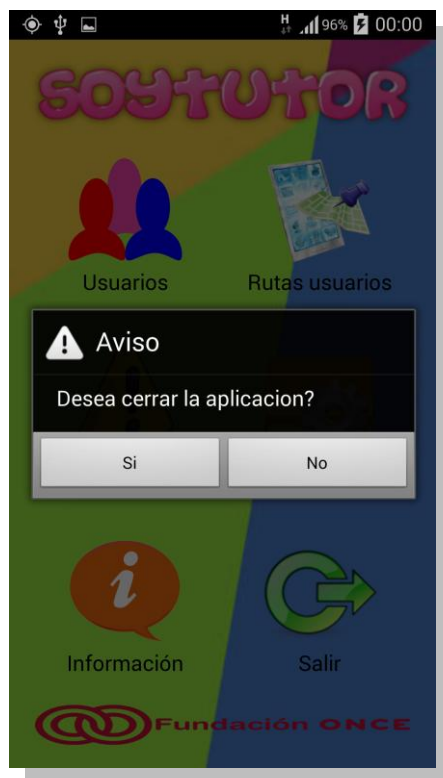
El usuario podrá modificar el teléfono que utilizara para los casos de emergencia, podrá ir a la configuración del GPS del móvil para activarla sino la tuviese y podrá desactivar o activar el envío de notificaciones.



*Ilustración 104: Manual Aplicación guía: Configuración.*

## Cerrar Aplicación

Utilizando esta opción el usuario cerrara la aplicación.



*Ilustración 105: Manual Aplicación guía: Cerrar aplicación.*



## Anexo 3: Registro de reuniones

**Fecha:** Jueves 10 de Octubre de 2013

**Hora:** 16:00

**Lugar:** Sala de reuniones de la facultad de informática

**Asistentes:** Victoria López, Jesús Escribano, Mario Arranz, Sergio Cabrerizo y Laura Aragón.

**Temas tratados:** Reunión inicial para ver la estructura e ideas iniciales del proyecto de la asignatura de sistemas informáticos.

**Fecha:** Jueves 31 de Octubre de 2013

**Hora:** 15:45

**Lugar:** Sala de reuniones de la facultad de informática

**Asistentes:** Victoria López, Jesús Escribano, Inés María Gómez, Mario Arranz, Sergio Cabrerizo y Laura Aragón.

**Temas tratados:** Definir presentación para mostrar en Fundación ONCE, estado del arte, primer prototipo de la página Web y definir funcionalidad de las aplicaciones móviles.

**Fecha:** Jueves 4 de Noviembre de 2013

**Hora:** 12:00

**Lugar:** Calle Sebastián Herrera, 15. Fundación ONCE

**Asistentes:** M<sup>a</sup> José Sánchez, David Zanoletty, Victoria López, Jesús Escribano, Mario Arranz, Sergio Cabrerizo y Laura Aragón.

**Temas tratados:** definir requisitos de la página web y de las aplicaciones móviles tanto para discapacitados psíquicos como para invidentes.

**Fecha:** Jueves 5 de Diciembre de 2013

**Hora:** 16:00

**Lugar:** Sala de reuniones de la facultad de informática

**Asistentes:** Victoria López, Jesús Escribano, Mario Arranz, Sergio Cabrerizo y Laura Aragón.

**Temas tratados:** mostrar segundo prototipo de la página web y añadir nueva funcionalidades.

**Fecha:** Jueves 16 de Enero de 2014

**Hora:** 16:00

**Lugar:** Sala de reuniones de la facultad de informática

**Asistentes:** Victoria López, Jesús Escribano, Mario Arranz, Sergio Cabrerizo y Laura Aragón.

**Temas tratados:** mostrar tercer prototipo de la página web, añadir nueva funcionalidades y mostrar primer prototipo de la aplicación móvil del usuario

**Fecha:** Jueves 20 de Febrero de 2014

**Hora:** 11:00

**Lugar:** Calle Sebastián Herrera, 15. Fundación ONCE

**Asistentes:** David Zanoletty, Victoria López, Jesús Escribano, Inés María Gómez, Mario Arranz, Sergio Cabrerizo y Laura Aragón.

**Temas tratados:** presentación de la página web, añadir nueva funcionalidades y definir requisitos de las aplicaciones móviles

**Fecha:** Miércoles 7 de Abril de 2014

**Hora:** 19:00

**Lugar:** Sala de reuniones de la facultad de informática

**Asistentes:** Victoria López, Mario Arranz, Sergio Cabrerizo y Laura Aragón.

**Temas tratados:** mostrar lo realizado hasta la fecha, página web, aplicación móvil del tutor y aplicación móvil del usuario

**Fecha:** Miércoles 29 de Abril de 2014

**Hora:** 19:00

**Lugar:** Sala de reuniones de la facultad de informática

**Asistentes:** Victoria López, Jesús Escribano, Mario Arranz, Sergio Cabrerizo y Laura Aragón.

**Temas tratados:** mostrar página web terminada, prototipos de casi acabadas de las aplicaciones móviles y definición de las partes de la memoria.



# Índice de figuras

Ilustración 1: Aplicación Accessibility: modos a elegir.....	3
Ilustración 2: Aplicación Accessibility: Realidad aumentada.....	4
Ilustración 3: Aplicación Consultar puntos de interés.....	4
Ilustración 4: Aplicación Accessibility: Añadir puntos de interés .....	6
Ilustración 5: Aplicación Accessibility: Herramientas.....	6
Ilustración 6: Aplicación SafetyGPS V: Pantalla principal.....	7
Ilustración 7: Aplicación SafetyGPS V: Funcionalidades.....	8
Ilustración 8: Aplicación Sloper: Representación de la ruta .....	9
Ilustración 9: Diagrama de tareas .....	23
Ilustración 10: Diagrama de Gantt .....	23
Ilustración 11: Diagrama de recursos .....	24
Ilustración 12: Diagrama de la base de datos .....	25
Ilustración 13: Diagrama general de casos de uso de la página Web.....	29
Ilustración 14: Diagrama de actividad del caso de uso Añadir Ruta de la página Web .....	32
Ilustración 15: Diagrama de actividad del caso de uso Añadir Usuario de la página Web.....	33
Ilustración 16: Diagrama de actividad del caso de uso Cerrar sesión de la página Web .....	34
Ilustración 17: Diagrama de actividad del caso de uso Eliminar Ruta de la página Web .....	35
Ilustración 18: Diagrama de actividad del caso de uso Eliminar Usuario de la página Web .....	37
Ilustración 19: Diagrama de actividad del caso de uso Exportar usuarios de la página Web .....	38
Ilustración 20: Diagrama de actividad del caso de uso Importar Usuarios de la página Web ....	39
Ilustración 21: Diagrama de actividad del caso de uso Inicio sesión de la página Web .....	40
Ilustración 22: Diagrama de actividad del caso de uso Menú configuración de la página Web ..	41
Ilustración 23: Diagrama de actividad del caso de uso Menú rutas de la página Web .....	42
Ilustración 24: Diagrama de actividad del caso de uso Menú usuarios de la página Web .....	43
Ilustración 25: Diagrama de actividad del caso de uso Modificar Ruta de la página Web .....	45
Ilustración 26: Diagrama de actividad del caso de uso Modificar Usuario de la página Web ....	47
Ilustración 27: Diagrama de actividad del caso de uso Mostrar rutas de un usuario de la página Web .....	48
Ilustración 28: Diagrama de actividad del caso de uso Registro de la página Web .....	49
Ilustración 29: Diagrama de actividad del caso de uso Valorar Ruta de la página Web.....	51
Ilustración 30: Diagrama de actividad del caso de uso Visualizar información Usuario de la página Web.....	52
Ilustración 31: Diagrama de actividad del caso de uso Visualizar Rutas de la página Web .....	53
Ilustración 32: Diagrama general de casos de uso de la aplicación del usuario.....	59
Ilustración 33: Diagrama de actividad del caso de uso cerrar sesión de la aplicación del usuario .....	60
Ilustración 34: Diagrama de actividad del caso de uso guiar por la ruta de la aplicación del usuario .....	62
Ilustración 35: Diagrama de actividad del caso de uso iniciar ruta de la aplicación del usuario	64
Ilustración 36: Diagrama de actividad del caso de uso iniciar sesión de la aplicación del usuario .....	66
Ilustración 37: Diagrama de actividad del caso de llamada de pánico de la aplicación del usuario .....	67

Ilustración 38: Diagrama de actividad del caso de uso Llamar tutor de la aplicación del usuario .....	68
Ilustración 39: Diagrama de actividad del caso de uso Ver configuración de la aplicación del usuario .....	69
Ilustración 40: Diagrama de actividad del caso de uso Marcar incidencia de la aplicación del usuario .....	71
Ilustración 41: Diagrama de actividad del caso de uso Ver información de la aplicación del usuario .....	72
Ilustración 42: Diagrama de actividad del caso de uso volver a la ruta de la aplicación del usuario .....	73
Ilustración 43: Diagrama general de casos de uso de la aplicación del guía.....	81
Ilustración 44: Diagrama de actividad del caso de uso Cerrar sesión de la aplicación del guía ..	82
Ilustración 45: Diagrama de actividad del caso de uso Inicio sesión de la aplicación del guía ..	84
Ilustración 46: Diagrama de actividad del caso de uso Llamar usuario de la aplicación del guía .....	85
Ilustración 47: Diagrama de actividad del caso de uso Ver configuración de la aplicación del guía.....	86
Ilustración 48: Diagrama de actividad del caso de uso Ver incidencia de la aplicación del guía	88
Ilustración 49: Diagrama de actividad del caso de uso Ver información del usuario de la aplicación del guía.....	90
Ilustración 50: Diagrama de actividad del caso de uso Ver información de la aplicación del guía .....	91
Ilustración 51: Diagrama de actividad del caso de uso Ver ruta de la aplicación del guía .....	93
Ilustración 52: Manual Página web: Acceso. ....	109
Ilustración 53: Manual Página web: Registro. ....	110
Ilustración 54: Manual Página web: Inicio sesión.....	110
Ilustración 55: Manual Página web: Menú usuarios. ....	111
Ilustración 56: Manual Página web: Menú Rutas. ....	111
Ilustración 57: Manual Página web: Menú configuración. ....	111
Ilustración 58: Manual Página web: Añadir nuevo usuario. ....	112
Ilustración 59: Manual Página web: Elegir usuario a eliminar. ....	113
Ilustración 60: Manual Página web: Confirmar eliminar usuario seleccionado.....	113
Ilustración 61: Manual Página web: Seleccionar usuario a modificar. ....	114
Ilustración 62: Manual Página web: Modificar usuario seleccionado.....	114
Ilustración 63: Manual Página web: Visualizar información de un usuario. ....	115
Ilustración 64: Manual Página web: Seleccionar usuario para configurar la nueva ruta. ....	115
Ilustración 65: Manual Página web: Insertar nombre de la nueva ruta. ....	116
Ilustración 66: Manual Página web: Insertar inicio de la ruta.....	116
Ilustración 67: Manual Página web: Insertar final de la ruta.....	117
Ilustración 68: Manual Página web: Añadir nueva ruta. ....	117
Ilustración 69: Manual Página web: Eliminar ruta.....	118
Ilustración 70: Manual Página web: Visualizar ruta sin valorar. ....	118
Ilustración 71: Manual Página web: Visualizar ruta valorada.....	119
Ilustración 72: Manual Página web: Visualizar rutas de un usuario. ....	119
Ilustración 73: Manual Página web: Valorar una ruta.....	120
Ilustración 74: Manual Página web: Instrucciones de cómo construir una ruta.....	120
Ilustración 75: Manual Página web: Configurar una ruta a valorar. ....	121
Ilustración 76: Manual Página web: Valorar tramos de la ruta por aceras.....	121

Ilustración 77: Manual Página web: Modificar nombre de la ruta.....	122
Ilustración 78: Manual Página web: Modificar valoración de los tramos de la ruta.....	122
Ilustración 79: Manual Página web: Cerrar sesión.....	123
Ilustración 80: Manual Aplicación usuario: Iniciar sesión.....	124
Ilustración 81: Manual Aplicación usuario: Selección usuario.....	124
Ilustración 82: Manual Aplicación usuario: Menú principal.....	125
Ilustración 83: Manual Aplicación usuario: Selección ruta.....	126
Ilustración 84: Manual Aplicación usuario: Activar GPS.....	126
Ilustración 85: Manual Aplicación usuario: Esperando GPS.....	127
Ilustración 86: Manual Aplicación usuario: Guiando al usuario.....	127
Ilustración 87: Manual Aplicación usuario: Volver a la ruta.....	128
Ilustración 88: Manual Aplicación usuario: Mapa incidencia.....	129
Ilustración 89: Manual Aplicación usuario: Mapa incidencia 2.....	129
Ilustración 90: Manual Aplicación usuario: Información.....	130
Ilustración 91: Manual Aplicación usuario: Configuración.....	131
Ilustración 92: Manual Aplicación usuario: Cerrar aplicación.....	132
Ilustración 93: Manual Aplicación guía: Inicio sesión.....	133
Ilustración 94: Manual Aplicación guía: Menú principal.....	134
Ilustración 95: Manual Aplicación guía: Selección usuario.....	135
Ilustración 96: Manual Aplicación guía: Datos del usuario.....	135
Ilustración 97: Manual Aplicación guía: Selección ruta.....	136
Ilustración 98: Manual Aplicación guía: Visualización ruta.....	136
Ilustración 99: Manual Aplicación guía: Lista incidencias.....	137
Ilustración 100: Manual Aplicación guía: Lista incidencias 2.....	137
Ilustración 101: Manual Aplicación guía: Visualización incidencia.....	138
Ilustración 102: Manual Aplicación guía: Notificación incidencia.....	138
Ilustración 103: Manual Aplicación guía: Información.....	139
Ilustración 104: Manual Aplicación guía: Configuración.....	140
Ilustración 105: Manual Aplicación guía: Cerrar aplicación.....	141



# Índice de tablas

Tabla 1: Caso de uso: Añadir ruta - Aplicación WebTutor .....	30
Tabla 2: Caso de uso: Añadir Usuario - Aplicación WebTutor .....	33
Tabla 3: Caso de uso: Cerrar Sesión - Aplicación WebTutor .....	34
Tabla 4: Caso de uso: Eliminar Ruta - Aplicación WebTutor .....	35
Tabla 5: Caso de uso: Eliminar Usuario - Aplicación WebTutor .....	36
Tabla 6: Caso de uso: Exportar Usuarios - Aplicación WebTutor .....	38
Tabla 7: Caso de uso: Importar Usuarios - Aplicación WebTutor .....	39
Tabla 8: Caso de uso: Inicio Sesión - Aplicación WebTutor .....	40
Tabla 9: Caso de uso: Menú Configuración - Aplicación WebTutor .....	41
Tabla 10: Caso de uso: Menú Rutas - Aplicación WebTutor .....	42
Tabla 11: Caso de uso: Menú usuarios - Aplicación WebTutor .....	43
Tabla 12: Caso de uso: Modificar ruta - Aplicación WebTutor .....	44
Tabla 13: Caso de uso: Modificar Usuario - Aplicación WebTutor .....	46
Tabla 14: Caso de uso: Modificar Rutas De Un Usuario - Aplicación WebTutor .....	48
Tabla 15: Caso de uso: Registro - Aplicación WebTutor .....	49
Tabla 16: Caso de uso: Valorar Ruta - Aplicación WebTutor .....	50
Tabla 17: Caso de uso: Visualizar Información de Un Usuario - Aplicación WebTutor .....	52
Tabla 18: Caso de uso: visualizar Rutas - Aplicación WebTutor .....	53
Tabla 19: Caso de uso: Cerrar Sesión - Aplicación MiTutor .....	60
Tabla 20: Caso de uso: Guiar Por La Ruta - Aplicación MiTutor .....	61
Tabla 21: Caso de uso: Iniciar ruta - Aplicación MiTutor .....	63
Tabla 22: Caso de uso: Iniciar Sesión - Aplicación MiTutor .....	65
Tabla 23: Caso de uso: Llamada De Pánico - Aplicación MiTutor .....	67
Tabla 24: Caso de uso: Llamar Tutor - Aplicación MiTutor .....	68
Tabla 25: Caso de uso: Ver Configuración - Aplicación MiTutor .....	69
Tabla 26: Caso de uso: Marcar Incidencia - Aplicación MiTutor .....	70
Tabla 27: Caso de uso: Ver Información - Aplicación MiTutor .....	72
Tabla 28: Caso de uso: Volver A La Ruta - Aplicación SoyTutor .....	73
Tabla 29: Caso de uso: Cerrar Sesión - Aplicación SoyTutor .....	82
Tabla 30: Caso de uso: Inicio Sesión - Aplicación SoyTutor .....	83
Tabla 31: Caso de uso: Llamar Usuario - Aplicación SoyTutor .....	85
Tabla 32: Caso de uso: Ver Configuración - Aplicación SoyTutor .....	86
Tabla 33: Caso de uso: Ver Incidencia - Aplicación SoyTutor .....	87
Tabla 34: Caso de uso: Ver Información Usuario - Aplicación SoyTutor .....	89
Tabla 35: Caso de uso: Ver Información - Aplicación SoyTutor .....	91
Tabla 36: Caso de uso: Ver Ruta - Aplicación SoyTutor .....	92